Uma Arquitetura para Análise e Visualização de Tráfego de Rede Malicioso

Heitor Ricardo Alves de Siqueira[1,2], Alexandre Or Cansian Baruque[1,2], Paulo Lício de Geus[2], André Ricardo Abed Grégio[1,2]

[1] Centro de Tecnologia da Informação Renato Archer (CTI/MCTI)
Rod. D. Pedro I (SP-65), KM 143,6 – 13069-901 – Campinas – SP – Brasil

[2]Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
Av. Albert Einstein, 1251 – 13083-852 – Campinas – SP – Brasil

{heitor,baruque,paulo}@lasca.ic.unicamp.br, andre.gregio@cti.gov.br

Abstract. *The Internet is an effective vector to spread attacks and malware. While exploits and malicious codes search for and attack vulnerable systems through the Internet, those systems already infected by malware use the network to send information obtained from the victim. Thus, there is a need for the analysis of network traffic captured during the interaction between malware and their targets: honeypots or dynamic malware analysis systems. In this work, we introduce an architecture to extract information from suspicious network traffic, as well as a Web interface for the visualization of traffic features. We obtained results from over 3 thousand network traffic files produced by malware and from over 2 million honeypot connections.*

Resumo. *A Internet é um veículo eficaz para a disseminação de ataques e programas maliciosos. Enquanto códigos de exploração buscam e atacam sistemas vulneráveis via Internet, sistemas já infectados por malware usam a rede para enviar informações obtidas da vítima. Com isso, há a necessidade de se analisar o tráfego de rede capturado durante a interação de programas maliciosos com o alvo, seja ele um honeypot ou um sistema de execução dinâmica de malware. Neste trabalho, introduz-se uma arquitetura para extrair informações de tráfego de rede suspeito e uma interface Web para visualização de características deste tráfego. Os resultados foram obtidos de mais de 3 mil arquivos de tráfego gerados por malware e de mais de 2 milhões de conexões a honeypots.*

1. Introdução

Uma das formas mais comuns para a disseminação de programas maliciosos é através da Internet. Programas maliciosos—*malware*—são, em geral, nomeados de acordo com o comportamento principal que apresentam [Skoudis and Zeltser 2003]. Por exemplo, os *worms* são classes de *malware* que se propagam autonomamente e exploram vulnerabilidades dos alvos; *downloaders* obtêm módulos maliciosos via rede; *stealers* roubam informações da vítima e as enviam para um servidor remoto. O ponto comum entre as classes mencionadas é o uso da rede para alcançar o ataque, seja para propagação e obtenção de *malware*, ou para exfiltração de dados sensíveis.

A coleta de exemplares de *malware* para análise pode ser feita de várias maneiras, tais como *crawling* de URLs (*Uniform Resource Locator*) em mensagens de e-mail contaminadas, *honeypots*[1] específicos, como a `Dionaea`[2] (emulador de serviços de rede vulneráveis do Windows), etc. A execução dos exemplares coletados, por sua vez, pode ser feita com sistemas de análise dinâmica publicamente disponíveis, como Anubis [Kruegel et al. 2006, iSecLab 2015] e Cuckoo Sandbox [Guarnieri et al. 2015]. Ao fim da análise dinâmica, tem-se um registro das atividades efetuadas pelo *malware* no alvo, bem como o tráfego de rede gerado durante a infecção.

Os objetivos deste trabalho são (i) investigar o tráfego de rede capturado durante a execução de *malware* em sistemas de análise dinâmica, bem como dados de rede provenientes da exploração de alguns *honeypots* de coleta de exemplares, e (ii) visualizar informações resultantes dessa investigação. Para tanto, propõe-se uma arquitetura de extração de dados e visualização de informações de tráfego de rede baseada em ferramentas livres e no desenvolvimento de um *framework* Web próprio. Com isso, espera-se reconhecer padrões de ataque e tendências de atuação de programas maliciosos, de modo a permitir a correlação de dados para identificar pontos de obtenção de módulos suspeitos ou de envio de informações da vítima. O restante deste artigo está dividido como segue: na Seção 2 são apresentados alguns trabalhos relacionados; na Seção 3, descreve-se a arquitetura e operação do *framework* desenvolvido para processamento do tráfego de rede capturado e visualização dos dados analisados; a Seção 4 mostra os resultados obtidos de tráfego coletado via execução de *malware* e em *honeypots* distribuídos; por fim, a Seção 5 apresenta as considerações finais.

2. Trabalhos Relacionados

[Hoepers et al. 2005] introduzem uma infraestrutura para detecção de ameaças e coleta de dados fazendo se uso de *honeypots* distribuídas e uma central para a agregação e processamento dos dados coletados. Foi feito o uso de *honeypots* de baixa interação (*honeyd*) para a coleta de dados, os quais estão disponíveis publicamente em *honeyTARG*[3]. Neste artigo, faz-se uso desses dados para efeitos de comparação quando cabível.

[Ceron et al. 2012] mostram uma análise detalhada sobre os ataques a servidores que utilizam o protocolo SIP (*Session Initiation Protocol*). Este tipo de análise é importante devido ao número crescente de serviços VoIP (*Voice over IP*)providos pela rede e para que se possa entender o tipo de ataque efetuado contra o protocolo e as vulnerabilidades comumente exploradas. Os *honeypots* utilizados no presente artigo também simulam um serviço de VoIP para coletar dados cujas estatísticas são apresentadas adiante.

[Rossow et al. 2011] apresentam uma análise em larga escala sobre a atividade em rede de amostras de *malware* do tipo "*downloader*". Foi feita uma inspeção em baixo nível dos protocolos DNS (*Domain Name System*), HTTP (*Hypertext Transfer Protocol*), IRC (*Internet Relay Chat*) e SMTP (*Simple Mail Transfer Protocol*). Como relatado, os protocolos DNS e HTTP ocorrem com uma frequência significativamente maior que outros, reforçando a importância de seu estudo no contexto de inspeção de atividades ma-

[1] *Honeypots* são recursos computacionais que visam ser comprometidos para estudo dos métodos, ferramentas e motivações dos atacantes [Spitzner 2003].

[2] `http://dionaea.carnivore.it/`

[3] `http://honeytarg.cert.br/`

liciosas. Apesar desse trabalho apresentar uma análise extremamente detalhada de tráfego malicioso, a inspeção é feita apenas no conteúdo dos pacotes capturados pelos protocolos DNS e HTTP. O artigo atual faz inspeção de ainda outros protocolos de aplicação, apresentando também um modo mais visual de interpretar as informações analisadas.

O artigo [Filho et al. 2010] apresenta uma ferramenta para análise dinâmica de *malware* através da monitoração de chamadas de sistema e tráfego de rede capturado. Apesar de ter um foco no estudo comportamental de código malicioso, o sistema mostrado nesse artigo auxiliou na obtenção de amostras de tráfego garantidamente maliciosos. Várias amostras provenientes da execução de *malware* foram utilizadas no projeto atual, o que forneceu estatísticas valiosas quanto à atividade em rede de programas maliciosos atuantes no Brasil.

3. Arquitetura para Investigação de Tráfego Malicioso

A arquitetura projetada para analisar informações do tráfego de rede capturado é composta por dois componentes principais: um sistema responsável pela extração e processamento dos dados; um *framework* Web para visualização dinâmica das informações processadas. Tal arquitetura está ilustrada na Figura 1.

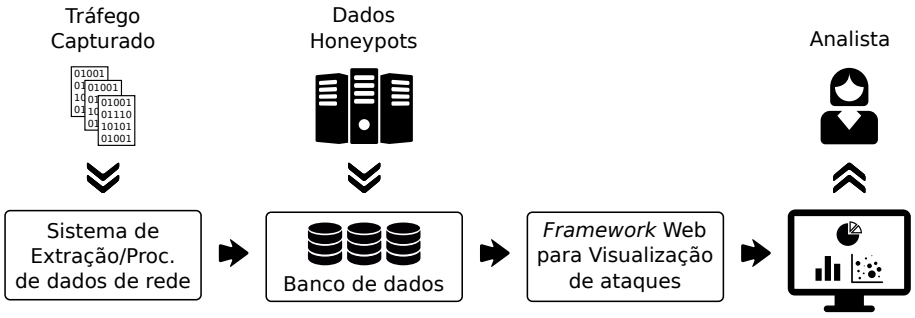

Figura 1. Esquematização simplificada da arquitetura proposta.

3.1. Coleta de Dados

Para os resultados deste artigo, trabalhou-se com duas fontes de dados:

- A primeira delas provém da análise dinâmica de exemplares de *malware* coletados no período de novembro/2013 até maio/2014. Os exemplares foram executados durante aproximadamente cinco minutos em ambiente controlado e todo o tráfego de rede gerado durante a execução de cada exemplar foi capturado e armazenado;
- A segunda fonte de dados consiste de exemplares de *malware* coletados por uma rede composta por *honeypots* do tipo `Dionaea` distribuídos em quatro instituições brasileiras de ensino e pesquisa. A coleta via `Dionaea` ocorreu entre outubro/2014 e julho/2015.

O conjunto total de exemplares coletados nos períodos citados foi de 3.410 programas maliciosos, os quais foram analisados dinamicamente para gerar os *logs* de tráfego de rede no formato `PCAP` (*Packet Capture*) a serem investigados por meio da arquitetura proposta. Outras 2.288.235 conexões distintas aos *honeypots* disponíveis também foram utilizados para visualização e produção de estatísticas.

3.2. Análise do Tráfego

A análise do tráfego de rede foi feita com auxílio da ferramenta `bro` [Paxson 1999]. Ao contrário de monitores de rede convencionais, o `bro` não se utiliza apenas do conceito de "assinaturas". Todo tráfego é transformado em uma série de *eventos* em alto nível que podem representar um *login* por FTP (*File Transfer Protocol*), uma conexão a um *site*, ou a transferência de um arquivo. Tais eventos são tratados por um sistema de *scripts* que permite uma inspeção aprofundada de acordo com o objetivo do analista de rede. Esse sistema permite a identificação de ataques e outras atividades maliciosas de maneira mais intuitiva que uma inspeção manual.

Para a geração das estatísticas apresentadas neste artigo, foi criado um novo *script* que processa as informações relevantes ao trabalho realizado. Além de armazenar informações básicas, como origem e destino de cada conexão de rede, os arquivos presentes no tráfego foram extraídos, dado que são potencialmente maliciosos. A identificação de um arquivo "potencialmente malicioso" se deu com base na diferença entre a declaração do *MIME type* de cada arquivo e o que este realmente é. Os arquivos extraídos foram submetidos ao VirusTotal[4] para detecção por *engines* de antivírus.

Em relação aos protocolos de aplicação, inicialmente foi feita a coleta de requisições HTTP (métodos `GET` e `POST`). No cenário brasileiro, é comum a atuação de *malware* voltado para o roubo de credenciais de usuários de Internet Banking, cuja exfiltração de dados sensíveis é feita por meio de métodos HTTP [Grégio et al. 2013]. Uma inspeção mais detalhada de tráfego HTTP auxilia de maneira significativa a identificação de servidores comprometidos usados para armazenar informações de usuários ou que transferem exemplares de *malware* para as vítimas.

Além de inspecionar o tráfego HTTP, consultas DNS foram utilizadas para identificar atividades e domínios referentes a *pharming* e *phishing*. Nestes tipos de ataque, a vítima é levada a um domínio malicioso acreditando ter acessado algum serviço legítimo [Bin et al. 2010]. Esses ataques são muito utilizados na falsificação de sites bancários, podendo fazer com que a vítima ceda suas credenciais a um agente malicioso. De acordo com estatísticas nacionais recentes [CERT.br 2015], ataques de fraude estão entre os incidentes mais reportados ao órgão, com um aumento expressivo de 80% em 2014 quando comparado ao ano anterior.

Adicionalmente aos protocolos citados acima, o tráfego foi processado de forma a se obter informações referentes a tráfego FTP, SMTP, TLS (*Transport Layer Security*) e SSL (*Secure Sockets Layer*). Estatísticas do protocolo FTP podem fornecer padrões interessantes sobre comprometimento de credenciais e invasões mais diretas. Tentativas de *login* malsucedidas são armazenadas e facilitam a identificação de ataques de força-bruta. Mensagens de e-mail capturadas pelo protocolo SMTP podem indicar que um *malware* está comunicando a seu "proprietário" o sucesso de um ataque ou as informações sigilosas roubadas de um usuário, e também podem ser utilizadas para se descobrir indícios de envio de Spam. Por fim, a inspeção de tráfego TLS/SSL indica quando uma tentativa de estabelecer conexões seguras foi feita, e se esta tentativa foi bem-sucedida.

Todas as informações coletadas pelo *script* desenvolvido são armazenadas em um banco de dados SQL (*Structured Query Language*), separadas de acordo com o protocolo

[4] `http://virustotal.com/`

de aplicação. Uma tabela adicional relaciona o *hash* de cada arquivo extraído com o endereço IP (*Internet Protocol*) de sua conexão. Esta organização possibilita a recuperação mais eficiente de estatísticas relevantes, e auxilia a etapa de processamento de dados para visualização no *framework* Web.

Suplementando os dados provenientes da execução de *malware*, o tráfego capturado nos *honeypots* também foi utilizado para geração de estatísticas. Para cada ataque registrado, diversas informações foram coletadas no intuito de reconhecer padrões e tendências de ataques em rede. Essas informações incluem, dentre outras, o protocolo e a porta utilizados no ataque e um *timestamp* de sua ocorrência. Os dados coletados nesta etapa também foram armazenados no banco de dados SQL.

3.3. Visualização Geográfica

A visualização das atividades de rede dos exemplares de *malware* coletados e analisados dinamicamente tem o objetivo de facilitar a interpretação dos dados por um analista humano e permitir a observação de tendências com relação aos dias e horários dos ataques, tipos, e local de origem do ataque/malware (ou destino dos dados enviados da ou transferidos para a vítima).

O desenvolvimento deste *framework* baseou-se na utilização da biblioteca *GeoIP*[5] para estimar a posição geográfica do ponto de ataque. A partir dessa informação e definido um período de tempo de interesse, é possível gerar um mapa que mostra os locais de origem dos ataques detectados pelos *honeypots* e sua evolução ao longo do período determinado. O mapa permite observar os dados sobre os ataques em uma dimensão temporal, atualizando as informações de acordo com o horário no qual tais dados foram coletados pelos sensores. Dessa forma, pode-se observar as tendências dos tipos de ataques no decorrer do dia, analisando-se os protocolos, serviços e locais de origem.

A Figura 2 representa um mapa em um dado instante de tempo com dados de potenciais ataques, apresentados de acordo com a localização geográfica do endereço IP obtido. Outras informações tais como a cidade estimada por meio das coordenadas obtidas do endereço IP, o tipo do ataque (serviço alvo) e a porta de rede vulnerável que recebeu o ataque também podem ser visualizadas. Os marcadores no mapa variam de tamanho de acordo com a frequência dos ataques, e de cor dependendo do protocolo utilizado. Logo abaixo do mapa, pode-se notar uma barra de rolagem, que se movimenta de forma automática para mostrar as tendências de ataques ao longo do dia.

4. Testes e Resultados

Uma vez que a arquitetura proposta é composta por dois subsistemas que lidam com dados de naturezas distintas (tráfego de rede gerado pela análise de *malware* e dados de ataques coletados pelos *honeypots*), a inspeção deles foi feita de modo específico. Os procedimentos tomados e resultados obtidos são detalhados adiante.

4.1. Análise Dinâmica

Após o processamento (via *script* feito para o `bro`) do tráfego capturado pela análise dinâmica de *malware*, 175 binários executáveis do tipo PE32 foram corretamente extraídos,

[5] `https://dev.maxmind.com/geoip/`

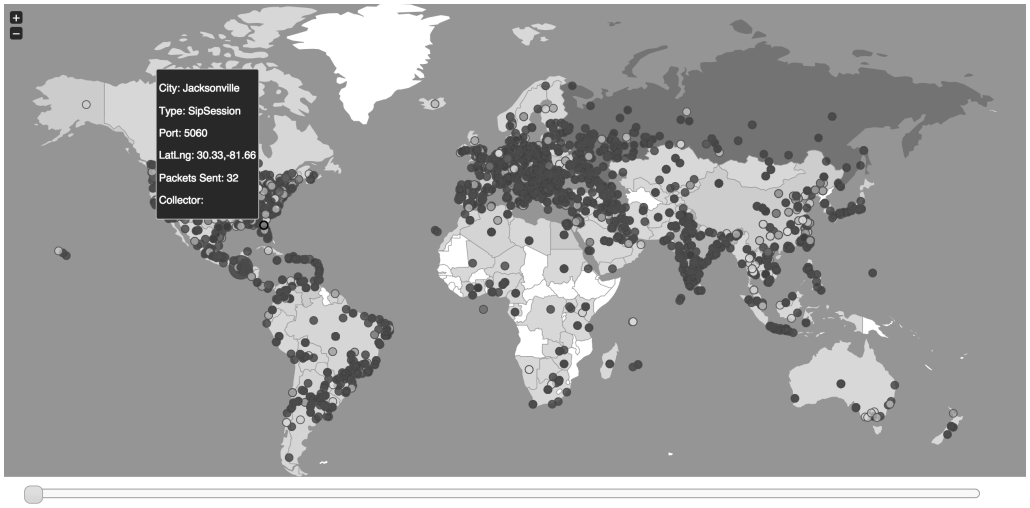

Figura 2. Mapa mostrado pela interface Web retratando os locais de origem dos ataques.

totalizando 250MB de tráfego. Estes arquivos foram submetidos ao VirusTotal para verificação, que detectou 103 deles como maliciosos (voltados para plataforma *Windows*). Assim, a porcentagem de arquivos sabidamente infectados é de 58,85%. Entretanto, o restante dos executáveis pode não ser benigno. Em relação ao tráfego HTTP, foram capturadas 14.828 requisições. Dessas, apenas 2.083 foram requisições do tipo `POST`. Na maioria dos casos, essas requisições faziam o *upload* de informações obtidas do sistema operacional da vítima por meios de *scripts* PHP. Logo, os exemplares de *malware* que efetuam esse tipo de tráfego podem ser considerados como *information stealers*. A maior parte do tráfego HTTP corresponde a requisições do tipo `GET` (12.475, ou 84,13% do tráfego HTTP capturado para análise). Portanto, esses exemplares podem ser considerados *downloaders*, dado que trazem algum arquivo para a máquina da vítima. 1.133 das conexões usando GET corresponderam especificamente ao *download* de executáveis.

A análise de tráfego DNS resultou em 5.357 requisições realizadas por esse protocolo. Em 4.692 delas, os domínios enviados para resolução não eram endereços válidos ou estavam indisponíveis, resultando em mensagens de erro. Apenas 665 requisições corresponderam a endereços resolvíveis, 354 dessas pedindo a resolução de domínios nacionais. Somente 11,76% das requisições DNS capturadas solicitam a resolução de domínios válidos em geral. Quanto a tentativas de comunicação segura através do uso dos protocolos TLS e SSL, foram identificadas 3.946 conexões. Dessas, 2.827 foram bem-sucedidas, uma porcentagem correspondente a 71,64%. Dentre todas as conexões seguras, a maior parte (3.863) utiliza o protocolo *TLSv1.0*, o que representa $\approx 98\%$ do total analisado. Os dados referentes a protocolos de aplicação podem ser conferidos na Tabela 1.

4.2. Dados Coletados pelos Honeypots

Na Figura 3, pode-se ver o total de conexões capturadas ao longo de todo o período de coleta. Observa-se uma taxa menor de coleta nas primeiras semanas (enquanto o sistema distribuído estava sendo colocado no ar e testado), e quedas na coleta em janeiro, março e alguns dias em abril. Isto ocorreu devido a problemas técnicos em alguns dos sensores, comprometendo a captura de dados.

Tabela 1. Relação de protocolos de aplicação no tráfego coletado da análise dinâmica de *malware*.

Protocolo	Número de Requisições	Porcentagem
HTTP	14.828	61,36%
DNS	5.357	22,17%
TLS/SSL	3.946	16,33%
FTP	23	0,09%
SMTP	12	0,05%
Total	24.166	100%

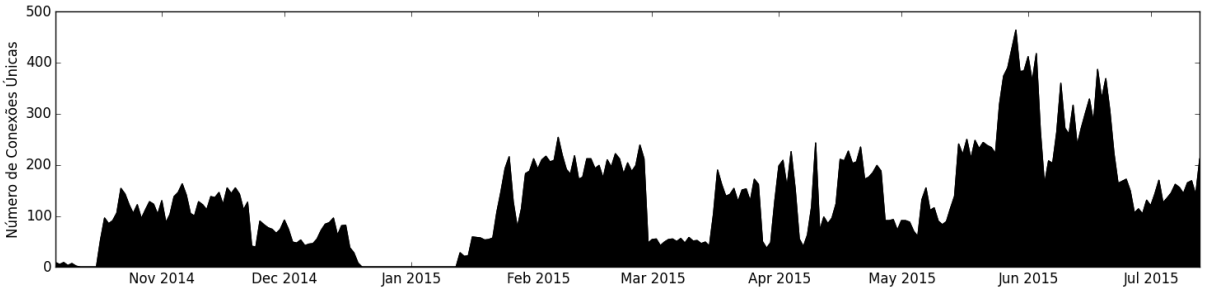

Figura 3. Volume de conexões únicas capturadas pelos *honeypots*.

Na Tabela 2, pode-se observar a quantidade de conexões de ataque capturadas pelos sensores. Os ataques foram separados por tipo de acordo com sua porta: FTP (21), HTTP (80), RPC (135, *Remote Procedure Call*), SMB (445, *Server Message Block*), MS-SQL (1433, *Microsoft SQL Server*) e SIP (5060, *Session Initiation Protocol*). A maioria dos ataques foram destinados aos serviços de SMB, HTTP e SIP.

Tabela 2. Estatísticas da coleta nos *honeypots* no período de 6 de Outubro de 2014 até 14 de Julho de 2015.

Porta	total de ataques	IPs únicos
21	654 (0,03%)	381 (1,49%)
80	31.001 (1,35%)	4.319 (16,94%)
135	369 (0,02%)	89 (0,35%)
445	2.153.774 (94,12%)	15.659 (61,41%)
1433	18.665 (0,82%)	1.292 (5,08%)
5060	83.771 (3,66%)	3.757 (14,73%)
Total	2.288.234 (100%)	25.497 (100%)

Na Figura 4, tem-se o gráfico da fração de endereços IP associados a atividades maliciosas que ficaram operacionais por um determinado número de dias. Extraindo alguns pontos desse gráfico, podemos ver que a grande maioria deles foram identificados em um único dia, 16,7% ficaram ativos por dois ou mais dias, 12,4% se mantiveram ativos por sete dias ou mais, 7,9% por mais de 30 dias, 4,3% por mais de 90 dias, e apenas 1,7% por mais de 180 dias. Houve também um número limitado de endereços IP (0,5%) que permaneceram em funcionamento durante todo o período de coleta. A Figura 5 mostra uma tendência dos ataques em ocorrer entre 10 e 16 horas.

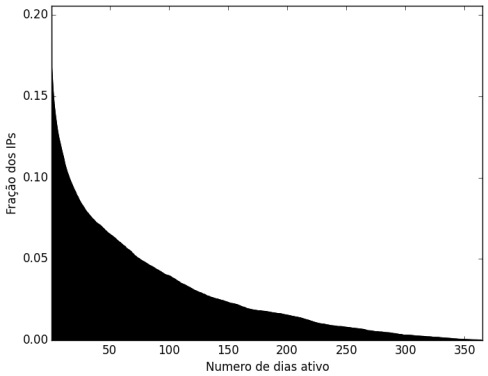

Figura 4. Distribuição do número de IPs ativos por dia em escala log.

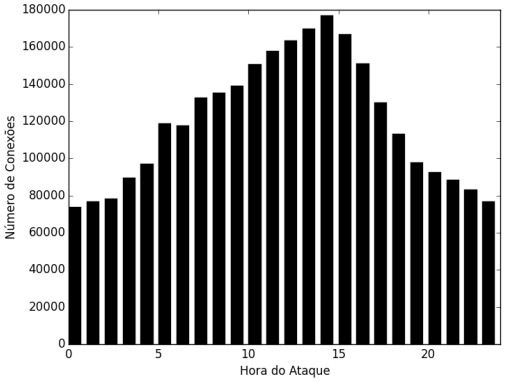

Figura 5. Distribuição dos ataques por hora (geral).

Para obter mais informações, foi gerado e analisado o mesmo tipo de histograma para cada tipo de ataque em separado. Na Figura 6 pode-se observar que essa variação no histograma geral tem sua origem no período de tempo durante o qual os ataques SMB ocorrem. Essa mesma tendência dos ataques contra a porta 445 também pode ser observada nas estatísticas produzidas pelo projeto *honeyTARG*, do CERT.br. Nos ataques contra FTP, observa-se que a maior parte das conexões ocorre entre 21 e 23 horas. Esse comportamento é ilustrado na Figura 7.

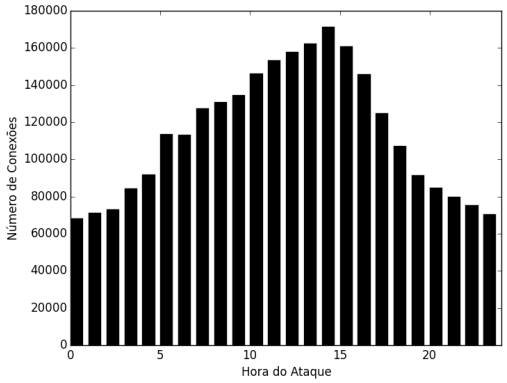

Figura 6. Distribuição dos ataques contra a porta 445 por hora.

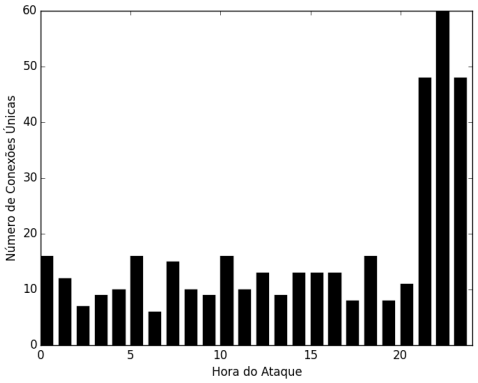

Figura 7. Distribuição dos ataques contra a porta 21 por hora.

Quanto aos ataques contra MSSQL, pode-se notar na Figura 8 que há um pico de conexões elevado às 13 horas. Ao comparar com a Figura 9, na qual foram desconsideradas as conexões repetidas de uma mesma origem, esse evento desaparece. Isto leva à inferência de que houve um ataque geral contra o serviço citado que reverberou nos *honeypots* utilizados neste artigo. Tal ataque, embora massivo do ponto de vista dos sensores utilizados (haja vista o pico observado), foi perpetrado por poucos endereços de origem.

5. Conclusão

Neste artigo, apresentou-se uma proposta de arquitetura para processamento e visualização de dados de tráfego de rede. Tal arquitetura é composta por dois subsistemas—um

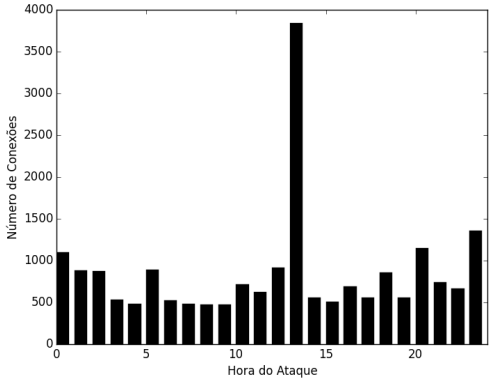

Figura 8. Distribuição dos ataques recebidos na porta 1433 (MSSQL) por hora.

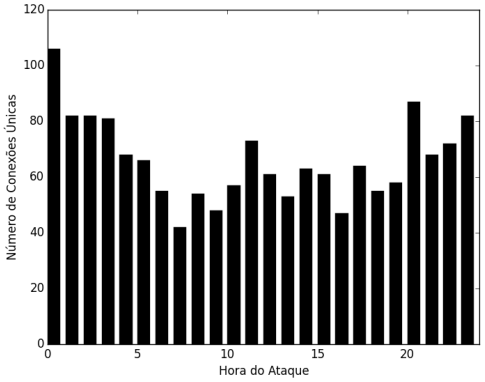

Figura 9. Distribuição dos ataques únicos contra a porta 1433 por hora.

para extração de dados em arquivos de tráfego de rede baseado em desenvolvimento de *script* para o sistema de detecção de intrusão `bro`, outro desenvolvido sob a forma de um *framework* Web para visualização geográfica dinâmica de ataques coletados por *honeypots*—e permite a análise de informações de rede em busca de tendências de ataques e atividades realizadas por programas maliciosos. Pôde-se observar que a maior parte do tráfego produzido por *malware* em execução se utilizou do método `GET` do protocolo HTTP para obtenção de componentes para a máquina infectada. Já nos *honeypots*, o serviço vulnerável mais atacado foi o SMB, na porta 445. Esse serviço é comumente explorado por *worms* há muitos anos, visando infectar máquinas expostas pelo serviço de compartilhamento de arquivos.

As possibilidades abertas durante o desenvolvimento desse trabalho incluem a automatização no envio de amostras de *malware* coletadas pelos *honeypots* para um sistema de análise dinâmica, visando a realimentação do sistema e provisão de informações adicionais. Com isso, é possível fazer a correlação entre ataques direcionados a *honeypots* e tráfego de rede proveniente da execução de *malware*, coletados nos *honeypots* ou não. Devido a cada subsistema parte da arquitetura ter ficado pronto em tempos diferentes e à própria natureza dos ataques e códigos maliciosos, não houve a possibilidade de se correlacionar os resultados das duas fontes de dados.

Portanto, o trabalho futuro natural, agora que a coleta de ambas as fontes (análise dinâmica e *honeypots*) está sendo feita constantemente e ao mesmo tempo, é a análise e o correlacionamento desses dados de forma a se identificar pontos de rede em comum para *download* de *malware* e exfiltração de informações sensíveis. Um trabalho já em desenvolvimento é uma ferramenta de geração e comparação de grafos, a qual será responsável por auxiliar na análise conjunta e visualização das operações realizadas nos diversos tráfegos e dados de rede capturados. Com isso, será possível a uma analista de segurança tomar decisões de gerenciamento de redes que possam minimizar danos em ataques por *malware* e evitar proliferações de infecções em outros sistemas monitorados.

6. Agradecimentos

Os autores agradecem o apoio recebido do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq via Projeto MCTI/CNPq/Universal-A 14/2014 (Processo 444487/2014-0) e bolsas de iniciação científica/tecnológica de ambos os estudantes.

Referências

Bin, S., Qiaoyan, W., and Xiaoying, L. (2010). A DNS based anti-phishing approach. In *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, volume 2, pages 262–265. IEEE.

Ceron, J. M., Steding-Jessen, K., and Hoepers, C. (2012). Anatomy of SIP attacks. *login, the USENIX magazine*, 37(6).

CERT.br (2015). Estatísticas dos incidentes reportados, disponível em `http://www.cert.br/stats/incidentes/`.

Filho, D. S. F., Grégio, A. R., Afonso, V. M., Santos, M. J., and de Geus, P. L. (2010). Análise comportamental de código malicioso através da monitoração de chamadas de sistema e tráfego de rede. *X Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*.

Grégio, A. R. A., Fernandes, D. S., Afonso, V. M., de Geus, P. L., Martins, V. F., and Jino, M. (2013). An empirical analysis of malicious internet banking software behavior. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1830–1835. ACM.

Guarnieri, C., Tanasi, A., Bremer, J., and Schloesser, M. (2015). The cuckoo sandbox, disponível em `http://www.cuckoosandbox.org/`.

Hoepers, C., Steding-Jessen, K., Cordeiro, L., and Chaves, M. (2005). A National Early Warning Capability Based on a Network of Distributed Honeypots. *17th Annual FIRST Conference on Computer Security Incident Handling, Singapore*, pages 2–5.

iSecLab (2015). Anubis – malware analysis for unknown binaries, disponível em `https://anubis.iseclab.org/`.

Kruegel, C., Kirda, E., and Bayer, U. (2006). Ttanalyze: A tool for analyzing malware. In *Proceedings of the 15th European Institute for Computer Antivirus Research (EICAR 2006) Annual Conference*. Best Paper Award.

Paxson, V. (1999). Bro: a System for Detecting Network Intruders in Real-Time. *Computer Networks*, 31(23-24):2435–2463.

Rossow, C., Dietrich, C. J., Bos, H., Cavallaro, L., Van Steen, M., Freiling, F. C., and Pohlmann, N. (2011). Sandnet: Network traffic analysis of malicious software. In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, pages 78–88. ACM.

Skoudis, E. and Zeltser, L. (2003). *Malware: Fighting Malicious Code*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Spitzner, L. (2003). Honeypots: catching the insider threat. In *19th Annual Computer Security Applications Conference, 2003. Proceedings.*, volume 32, pages 170–179. Ieee.