Ahead of the Curve: A Deeper Understanding of Network Threats Through Machine Learning

Joy Nathalie Avelino
Core Technology
Trend Micro Incorporated
Pasig City, Philippines
Joy_Avelino@trendmicro.com

Carmi Anne Loren Mora
Core Technology
Trend Micro Incorporated
Pasig City, Philippines
Carmi_Mora@trendmicro.com

Jessica Patricia Balaquit
Core Technology
Trend Micro Incorporated
Pasig City, Philippines
Jessica_Balaquit@trendmicro.com

***Abstract*—The role of big data and machine intelligence in the field of information security is gaining importance as malicious attackers use evasion techniques (polymorphism, encryption, obfuscation) to bypass signature-based detection. As most threats propagate through the network, it is important to have proactive techniques to discover an infection before it damages a computer.**

This paper will examine how header-based information as well as other characteristics in the HTTP network traffic can be used to train a machine learning model to capture malicious behavior.

Network streams tagged as malicious are preprocessed and clustered. It has been found that features in the raw byte stream augmented with handcrafted features are useful in learning the characteristics of network threats.

In specific clusters formed, it is possible to identify certain threats targeting a specific server, or if there are characteristics that can be observed in the injected code for exploit detection.

Clustering malicious network traffic leads to a better understanding of protection against these types of threats, identification of connected malware campaigns, and insight on future trends.

***Index Terms*—Computational Intelligence, Machine Learning, Big Data Applications, Information Security, Network Security, Intrusion Detection**

I. Introduction

A network intrusion attack refers to any compromise in the stability or security of information stored on connected computers [1]. There are many intrusion detection techniques and methods used for detecting network anomalies. The traditional method is to monitor network protocols using signature/behavior-based rules and heuristics [1] [2] [3].

Since not all malicious network traffic happens in post-infection, all attack phases are monitored from pre-infection to post-infection. This technique helps widen the discovery of current threats and can be used as a source of reference on how the threat behaves when an attack takes place [4].

Other techniques include: use of custom sandbox analysis and threat intelligence sharing. Even if a file was not detected through the network or in the system, the custom sandbox analysis enables one to discover advanced threats. Threat intelligence sharing, on the other hand, enables other security products to quickly contain the threat and prevent further attack.

While the described technologies can address most network threats, there are still some caveats [5].

Signature-based detection lacks flexibility; if the detected network traffic has a minor change (e.g., the spoofed header has a dynamic, randomly-generated Uniform Resource Identifier (URI)), the signature cannot detect it unless the signature is modified [2] [6]. In addition to this, the growing automation of attacks and the sheer amount of attacks makes manual inspection by analysts time-consuming [7]. Advanced threats can be covered by behavior-based or heuristic rules, however, these have the potential to be aggressive, and can thus lead to false positives [6] [8].

Is there a solution to solve the lack of flexibility for signature-based detection and the aggressiveness of behavior-based detection and heuristics? Yes, through machine learning [9].

Machine learning can process data beyond what humans can in a short span of time and evolve according to the data instances given as input. The data fed into the machine learning model are from in-the-wild data and thus, it is possible to obtain insights which aid in identifying targeted attacks and advanced threats.

The paper is organized as follows: the next section will define network flow data, machine learning, and the dataset used. The implementation of the clustering model is discussed in Section III. Section IV contains the evaluation of the model and the last section states the conclusion and future work.

II. Assumptions and Preliminaries

A. Network Flow Data

A flow is defined as a ”unidirectional stream of Internet Protocol (IP) packets that share a set of common properties: typically, the IP-five-tuple of protocol, source and destination IP addresses, source and destination flows” [10]. Flow data exported by a packet sniffer to a packet capture (PCAP) contains information that is useful for examining the traffic composition of different applications and services in the network. Its intent

978-1-5386-5457-6/18/$31.00 ©2018 IEEE

is not to steal information, but to help secure the network [11]. It can be used for the discovery and analysis of different kinds of network anomalies, such as targeted attacks or presence of botnets.

B. Machine Learning

There have been various studies in machine learning for network data, and a standard approach is to use machine learning to classify malicious from legitimate traffic [12] [13] [14]. In classification, or supervised learning, a significant amount of time and resources is needed to sift through the data and label it.

Most of the data encountered in the real-world is unlabeled. The benefits of using unlabelled data in research is the ability to find patterns and discover new relationships between data through clustering. Unsupervised learning is the method primarily used to explore unlabelled data. For new types of threats, clustering can be applied at real-time.

The method that will be used in this study is semi-supervised learning. This reduces the effort in labeling by processing a large amount of unlabeled data through clustering. The labels will be used for finding relationships between different malware families and how they differ from one another.

C. Dataset

It has been identified that a critical component in prior research utilizing machine learning on malicious network flow data is the dataset. Previous studies used existing public datasets [1] [15] or datasets generated in a controlled environment [12] [16].

This study utilizes in-the-wild network data that potentially contains new threats that have not been seen. The dataset used is from PCAPs of recent threats tagged as malicious by Trend Micro's network detection engine. The network flows that will be processed are Hypertext Transfer Protocol (HTTP) traffic, since HTTP is commonly used as medium for malicious activity [17]. The goal for this study is to get further information from the clustering results in order to provide timely and relevant coverage of the network threat landscape.

III. Implementation

A. Data Preprocessing

In a PCAP, malicious flows are often mixed with normal flows, thus making it susceptible to noise. The large volume of network data present indicates manual clean-up is resource-intensive. However, to ensure the clusters are representative of the current threat landscape, the collected data should be filtered as much as possible [18].

The method used in this study to mitigate the noise is to separate the capture into multiple streams, with each one considered as an individual data point. In addition to this, given that this study deals with malicious network flow, it is expected that non-standard headers and formatting aberrations may be found in the data and preprocessing should take these into account.

B. Feature Engineering

Stream headers and other relevant information have been used to generate the features fed into the clustering model. Some of the features used are taken from previous academic papers studying features for anomaly detection, such as the byte entropy, distribution, and standard deviation of the headers and payload [12] [19]. Care has been taken to abstract from concrete attack instances, as reliance on these would overfit to the malware present in the dataset and would prevent the model from fitting well to novel attack instances [20].

The features in this study are crafted to reflect the subject matter expertise of network threat detection experts, and discriminates between certain types of malware. Some only target specific server types, while others manifest characteristics that can hint on the kind of malicious content being delivered to a machine.

While there is great potential for machine learning to be applied in security research, it is challenging to translate network traffic to a format acceptable as input to a machine learning model [14] [21]. There are trade-offs in considering an ad-hoc versus an automated approach. For this study, both methods of feature engineering are applied. The features selected have undergone scaling and normalization before being fed to the clustering model.

C. Choosing the Clustering Algorithm

Three clustering implementations were considered to determine what is the ideal algorithm to use in this type of problem: k-means, Density-based Spatial Clustering of Applications with Noise (DBSCAN), and Hierarchical DBSCAN (HDBSCAN). The Python language is used to implement it, with k-means and DBSCAN from the scikit-learn [22] library and the HDBSCAN from a standalone library by McInnes et al. [23]

In k-means, clustering requires one to know the number of clusters beforehand. It may also output different results depending on where the initial point was placed - which makes the clustering it produces to be unstable. Since the number of clusters is unknown, there is a need for an algorithm that can estimate the number of clusters. In the threat landscape where new types of threats are continuously emerging, this model may have to be adjusted periodically.

Thus, only DBSCAN and HDBSCAN were used. In both density-based algorithms, the number of clusters is determined by its neighboring points. This solved the problem with setting the number of clusters whenever a new threat is discovered.

Final analysis was generated using the HDBSCAN algorithm. It extends DBSCAN by using a hierarchical approach before extracting the stable clusters. When compared to DBSCAN, the results produced by HDBSCAN is consistent with the understanding of the threats as reviewed by the domain experts. In semi-supervised learning, validation of the cluster still involves human factor. The hierarchical approach HDBSCAN employed is useful in understanding the results and helped augment human expertise - which will be illustrated in the next section.

IV. Results

Preliminary results in the utilization of the clustering model to cluster similar types of malicious network flows are promising. The following cluster visualization is produced using Embedding Projector [24].

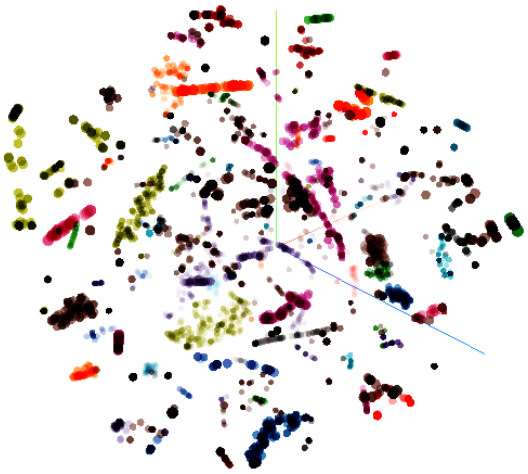

Fig. 1. Clusters formed by malicious HTTP streams. Each color represents one cluster

One of the clusters inspected predominantly has ransomware network flows. A ransomware is a type of malware which prevents or restricts users from accessing their system by locking the screen or encrypting their files until the user pays a ransom [25]. The Web plays an important part for ransomware because it requires a connection to the Command and Control (C&C) server to send an infection report, or receive the encryption key. Upon inspection of the ransomware cluster, most of the similarities occur in the Uniform Resource Locator (URL) found in various headers (URI, location, etc.)

Another interesting result comes from clusters comprised entirely of exploit kit detections (Figure 2). An exploit kit is type of toolkit cybercriminals use to exploit the vulnerabilities in a target system and perform Web-based attacks to propagate malware [26]. When these clusters were examined, the features most relevant to the clustering turned out to be those which concern file types. This makes sense - exploit kits can be differentiated by their target exploitable file format (Shockwave/Flash, PDF, JavaScript, etc.)

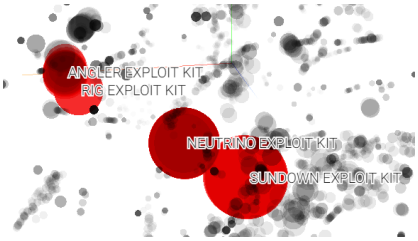

Fig. 2. Cluster that comprised different exploit kits

Figure 3 illustrates the different network characteristics for five malware families: Rig, FlashPack, Angler, Neutrino, and Blacole. All are exploit kits except for Blacole. The different colors correspond to structural attributes determined by the features passed to the model. For signature-based detection, one rule will be created for each family due to varying flow characteristics present in the network. But since signature-based detection lacks flexibility, having a slight change in the network traffic can render the rule unusable unless the signature is modified.

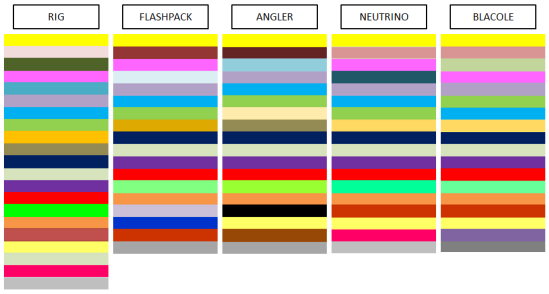

Fig. 3. Raw network data of each malware. Each color represents one characteristic.

Nevertheless, the clustering model was able to find similarities in the network flows, and group them together. From the multiple characteristics seen in each malware family, as illustrated in Figure 3, the clustering model was able to identify which ones constitute a certain profile that correlates among the similar samples. Figure 4 shows an analogy of how the clustering model sees the similar characteristics among the families.

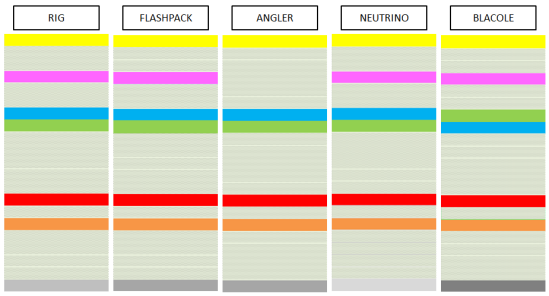

Fig. 4. Network flows as seen by the clustering model

Blacole, at first, seems like an outlier because it is a trojan and not an exploit kit. When its network traffic was investigated, it was found that its similarity with exploit kits is that it also takes advantage of the vulnerabilities present in JavaScript. This emphasizes the fact that new network traffic from exploit kits can be identified without tailoring the features to a specific attack instance.

The Ghost Remote Administration Tool (RAT) is a well-known trojan affiliated with GhostNet bot network and is commonly used in targeted attacks to gain control of the computer it infects [27]. Since the source code of GhostRAT is publicly available, a number of variants have emerged [28].

7hero	FWAPR	Heart	Level	QWPOT	X6RAT	xhjyk	cyl22	kaGni	Wh0vt
Adobe	FWKJG	IM007	Lover	Spidern	XDAPR	00000	DrAgOn	light	Snown
B1X6Z	GWRAT	ITore	Lyyyy	Tyjhu	Xjjhj	ABCDE	EXXMM	LkxCq	SocKt
BEiLa	Gh0st	KOBBX	MYFYB	URATU	aq0ft	apach	Eyes1	lvxYT	Super
BeiJi	GOLDt	KrisR	MoZhe	W0LFKO	attac	Assas	Gi0st	Naver	Sw@rd
ByShe	HEART	LUCKK	MyRat	Wangz	cb1st	Blues	GM110	NIGHT	v2010
FKJP3	HTTPS	LURK0	OXXMM	Winds	https	chevr	Hello	NoNul	VGTLS
FLYNN	HXWAN	LYRAT	PCRat	World	whmhl	CHINA	httpx	Origi	wcker
wings	X6M9K	xqwf7	YANGZ	QQ_124971919					

Fig. 5. Variants of GhostRAT [28]

As illustrated in Figure 6, the streams were clustered across multiple versions of GhostRAT because they contain similar payloads.

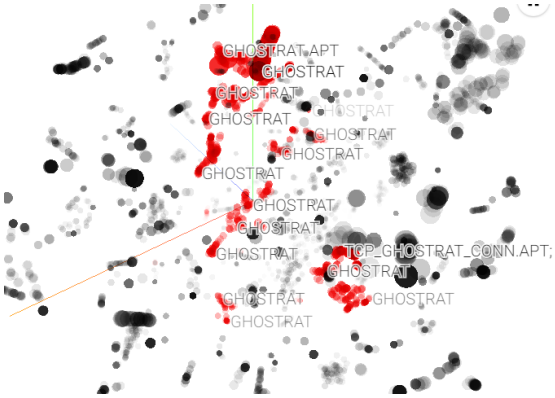

Fig. 6. GhostRAT Clusters

With threats reusing old malware to carry payload for backdoor capabilities (Figure 7), cryptocurrency mining (Figure 8) [29], and targeted attacks [30], machine learning can associate incoming traffic to future GhostRAT variants.

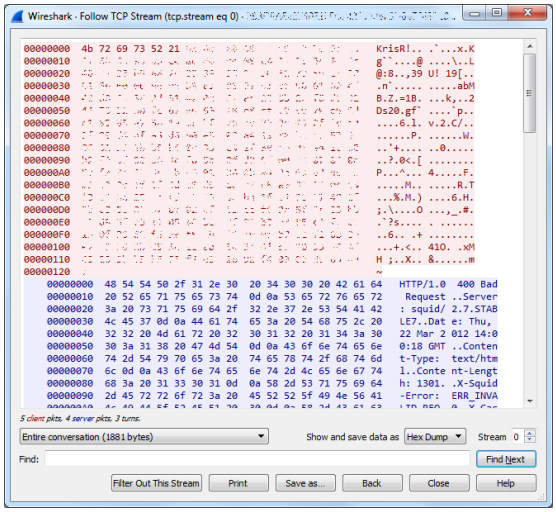

Fig. 7. Hex Dump of a GhostRAT Variant (KrisR)

V. Conclusion and Future Work

Clustering malicious network flows with features from the raw byte stream, augmented with handcrafted features as input from the data, can give insights on different network patterns from malicious traffic. It can also show similar characteristics

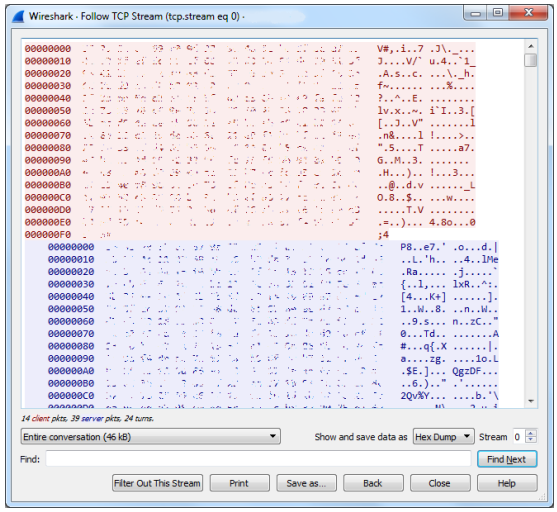

Fig. 8. Hex Dump of Monero Cryptocurrency Mining Payload

between different malware families, albeit within the same classification— such as exploit kits. This suggests that this approach is useful in augmenting signature creation for detecting network malware.

Using machine learning for analysis also vastly improves the speed at which data is organized and conclusions are obtained. The results show promise in utilizing machine learning to identify a widely-used vulnerability as it is spreading, or to recognize a certain vulnerability used in a novel way as part of another malware campaign.

Feature refinement can lead to better modelling of malicious flow clustering. At this stage, the model would benefit most from taking a closer look at URLs in the streams, and from other experimentation with other features extracted from the header contents, like measuring string randomness. For more accurate clustering of real-time detection, the model must be equipped, in future iterations, with the capacity to handle sequential data. This will also bolster its capabilities to cluster flows from other protocols than HTTP.

Acknowledgment

We would like to acknowledge the following people who gave their support in this project from Trend Micro Incorporated: Threat Research Director Mary Ong; Head of Machine Learning Group Brian Cayanan; Jameson Ong; machine learning consultants Jon Oliver, Abraham Camba, Jayson Pryde, Robert Tacbad, and Joa Suico; the Quality Assurance Team that pioneered this project; and the Deep Discovery Inspector Group that provided their threat expertise in network analysis.

References

[1] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*, Citeseer, 2001.

[2] G. Sanchez, "Don't always judge a packet by its cover," January 2016. White Paper. SANS Institute. [Online]. Available: https://www.sans.org/reading-room/whitepapers/access/dont-judge-packet-cover-36745.

[3] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, and K.-Y. Tung, "Review: Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, pp. 16–24, Jan. 2013.

[4] TrendMicro Incorporated, "Lateral movement: How Do Threat Actors Move Deeper Into Your Network?," July 2013. White Paper. TrendMicro Incorporated. [Online]. Available: http://about-threats.trendmicro.com/cloud-content/us/ent-primers/pdf/tlp_lateral_movement.pdf.

[5] TrendMicro Incorporated, "Inside the wire: Why Perimeter-centric Monitoring Leaves You Vulnerable," June 2015. White Paper. TrendMicro Incorporated. [Online]. Available: https://www.trendmicro.de/media/wp/ddi-network-vs-perimeter-wp-en.pdf.

[6] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, pp. 1–1, 2018.

[7] K. Rieck, "Computer security and machine learning: Worst enemies or best friends?," in *2011 First SysSec Workshop*, pp. 107–110, July 2011.

[8] C. Cade, "Understanding Heuristic-based Scanning vs. Sandboxing," 2015. https://www.opswat.com/blog/understanding-heuristic-based-scanning-vs-sandboxing.

[9] TrendMicro Incorporated, "There is no silver bullets: The strengths and weakness of todays threat-protection techniquesand why a multi-layered approach to endpoint security is a must," October 2016. White Paper. TrendMicro Incorporated. [Online]. Available: https://www.trendmicro.com/content/dam/trendmicro/global/en/business/products/user-protection/endpoint/wp_XGen-Silver-Bullet.pdf.

[10] G. Munz and G. Carle, "Real-time analysis of flow data for network attack detection," in *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, pp. 100–108, May 2007.

[11] M. A. Qadeer, A. Iqbal, M. Zahid, and M. R. Siddiqui, "Network traffic analysis and intrusion detection using packet sniffer," in *2010 Second International Conference on Communication Software and Networks*, pp. 313–317, Feb 2010.

[12] B. Anderson, S. Paul, and D. McGrew, "Deciphering malwares use of tls (without decryption)," *Journal of Computer Virology and Hacking Techniques*, pp. 1–17.

[13] D. Bekerman, B. Shapira, L. Rokach, and A. Bar, "Unknown malware detection using network traffic classification," in *2015 IEEE Conference on Communications and Network Security (CNS)*, pp. 134–142, Sept 2015.

[14] J. J. Davis, *Machine learning and feature engineering for computer network security*. PhD thesis, Queensland University of Technology, 2017.

[15] R. P. Lippmann, D. J. Fried, I. Graf, J. W. Haines, K. R. Kendall, D. McClung, D. Weber, S. E. Webster, D. Wyschogrod, R. K. Cunningham, and M. A. Zissman, "Evaluating intrusion detection systems: the 1998 darpa off-line intrusion detection evaluation," in *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, vol. 2, pp. 12–26 vol.2, 2000.

[16] S. Zhao, M. Chandrashekar, Y. Lee, and D. Medhi, "Real-time network anomaly detection system using machine learning," in *2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)*, pp. 267–270, March 2015.

[17] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser analysis of web-based malware," in *Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots'07, (Berkeley, CA, USA), pp. 4–4, USENIX Association, 2007.

[18] TrendMicro Incorporated, "Machine learning and next-generation intrusion prevention system (ngips): Building a Smarter NGIPS," June 2017. White Paper. TrendMicro Incorporated. [Online]. Available: https://documents.trendmicro.com/assets/wp/WP01_Machine_Learning_170608US.pdf.

[19] J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *computers & security*, vol. 30, no. 6-7, pp. 353–375, 2011.

[20] K. Rieck, "Machine learning for application-layer intrusion detection," 2009.

[21] H. Dong, J. Shang, D. Yu, and L. C. Lu, "Beyond the blacklists: Detecting malicious url through machine learning." BlackHat Asia, 2017.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, mar 2017.

[24] D. Smilkov, N. Thorat, C. Nicholson, E. Reif, F. B. Viégas, and M. Wattenberg, "Embedding projector: Interactive visualization and interpretation of embeddings," *arXiv preprint arXiv:1611.05469*, 2016.

[25] TrendMicro Incorporated, "What is Ransomware?," 2018. https://www.trendmicro.com/vinfo/ph/security/definition/ransomware.

[26] J. C. Chen and B. Li, "Evolution of exploit kits," *Trend Micro*, 2015.

[27] TrendMicro Incorporated, "Threat Encyclopedia: GHOSTRAT," 2018. https://www.trendmicro.com/vinfo/us/threat-encyclopedia/malware/ghostrat.

[28] S. Fagerland, "The many faces of gh0st rat: Plotting the connections between malware attacks," 2012. White Paper. Norman ASA. [Online]. Available: http://download01.norman.no/documents/ThemanyfacesofGh0stRat.pdf.

[29] N. Pantazopoulos, "Decoding network data from a Gh0st RAT variant," 2018. https://blog.trendmicro.com/trendlabs-security-intelligence/targeted-attack-in-taiwan-uses-infamous-gh0st-rat/.

[30] Z. Chang, K. Lu, A. Luo, C. Pernet, and J. Yaneza, "Operation iron tiger: Exploring Chinese Cyber-Espionage Attacks on United States Defense Contractors," July 2013. White Paper. TrendMicro Incorporated. [Online]. Available: http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-operation-iron-tiger.pdf.