Automated and Scalable Infrastructure for Hacker IRC Collection

Karan Chadha[1], Victor Benjamin[2], Youssif Al-Nashif[1], Salim Hariri[1], Hsinchun Chen[2]
[1]ECE Dept, [2]MIS Dept.
University of Arizona, Tucson, AZ USA
Contact Author: vabenji@email.arizona.edu

***Abstract*— Cyber security is an important challenge in today's world, as a growing amount of critical infrastructure has begun to rely on information technologies in order to meet increasingly complex demands. Traditional research in the cyber security domain has largely focused on improving security built directly into computing and networking systems. Conversely, little work has explored the human element behind cybercrime. . As a result, researchers and practitioners have taken an increased interest in advancing current cyber security capabilities by more closely examining hacker communities. However, hacker community data collection can often pose challenges that slow down or halt research progress. In this study, we design and implement a system that operationalizes the automated collection of hacker IRC contents. We detail challenges in data collection, as well as methods to circumvent such issues. We also summarize potential direction for future work, including the adoption of traditional analyses techniques to IRC data.**

Keywords— Cyber security; Hacker; Internet Relay Chat; IRC

I. INTRODUCTION

Cyber security is an important challenge in today's world, as a growing amount of critical infrastructure has begun to rely on information technologies in order to meet increasingly complex demands. Further, advancing technologies are enabling hackers to commit cybercrime at a much greater scale now than in the past. The growing number of emerging security threats necessitates further research and development for mitigating risk and exposure to vulnerabilities. As a result, researchers and practitioners have taken an increased interest in advancing current cyber security capabilities.

Traditional research in the cyber security domain has largely focused on improving security built directly into computing and networking systems. Typically, efforts are made in scrutinizing vulnerabilities at the protocol and system levels, where incremental advancements can be made in order to thwart existing security threats. Conversely, little work has explored the human element behind cybercrime; in particular, much is unknown concerning hacker behaviors, the cybercriminal supply chain, underground hacker communities, etc. Instead of taking a reactive approach to infrastructure protection and damage control, proactive cyber security attribution and situational awareness of the sources of attacks will allow researchers and practitioners to better understand the community of cyber attackers

In particular, methods to model cyber adversaries is one of the critical but unfulfilled research need outlined in a 2011 report on cyber security by the National Science and Technology Council [1]. Essentially, research on cybercriminals, would offer new perspectives for defending cyber space against Internet miscreants and actors with hostile intent. Further, recently published news reports detail how observation of hacker Internet-Relay-Chat (IRC) communities provided actionable information to cyber security professionals [2]. Analysis of hacker IRC data helped analysts uncover cybercriminal operations in motion, allowing for proactive actions against cyber-attack; specifically, analysts were able identify botnet operators, track the spread of malicious tools and malware, and identified key community participants.

However, despite the demonstrated importance of hacker IRC data, there appears to be a lack of research detailing procedures taken for the systematic identification, collection, and analysis of hacker IRC data. In this study, we design and implement a system that operationalizes the automated collection of hacker IRC contents. We detail challenges in data collection, as well as methods to circumvent such issues. We also summarize potential direction for future work, including the adoption of traditional analyses techniques to IRC data.

II. PREVIOUS WORKS

To form the basis for this research, literature is reviewed from two streams of research. First, we review previous hacker community research in order to better understand the context for this research. In particular, understanding more about the structure of hacker communities may aid in operationalizing the collection of hacker contents. Next, we review literature concerning past work on IRC data. There appears to be a stream of work investigating various IRC-based communities; such studies may provide insights for implementing collection mechanisms relevant to IRC.

A. Hacker Community Research

Hackers often congregate within online communities to learn, as hacking knowledge cannot be typically learned from formal institutions. In particular, hackers have been documented to use their communities for sharing various cybercriminal assets and hacking knowledge [3][4]. Thus, it is

likely that individuals with no prior hacking knowledge or skills may advance their capabilities by simply visiting hacker communities and learning from contents shared by other participants.

It appears that the majority of hacking communities scrutinized in past research are web forum communities or Internet Relay Chat (IRC) channels [3][4]. Methods to identify and collect such communities can be borrowed from previous studies. The most common approaches taken to identify hacker communities appear to focus on a combination of keyword searches as well as scrutinizing known communities for hyperlinks or references to other potential hacker communities [5][6].

Forum-based studies often highlight the common sharing of cybercriminal assets among community participants, and point out that such behavior is consistent in hacker communities across various geopolitical regions [3][7]. Participants of such forums may more closely collaborate or even conduct in black market transactions [8][9]. Results indicate that forums act as hubs for hackers to exchange assets and knowledge; continued research in this area will potentially provide greater understanding of hacker behaviors across geopolitical regions, information regarding the cybercriminal supply chain, growing threats, etc. Thus, close scrutiny of hacker communities may help support the overall goal of advancing current cyber security capabilities.

Unfortunately, a disproportionate amount of recent hacker community research scrutinizes forum data while largely ignoring IRC channel data. This may be due to several factors such. First, forums naturally create an archive of prior contents, allowing researchers to easily collect large amounts of data spanning multiple years. Conversely, IRC conversations occur in real-time and previous conversations are not normally archived; this requires real-time collection of data. Researchers would have to invest considerable effort ensuring complete collection of IRC data over time, while one can simply crawl web forum contents to potentially collect multiple years' worth of data. Additionally, many researchers may simply be unfamiliar with the IRC protocol and rather prefer the familiarity of webpage-based forums. Lastly IRC chat data is not indexed by common search engines, thus increasing the difficulty of identifying relevant conents.

The small amount of work that has been done on hacker IRC data identifies cybercriminal activity such as asset sharing and black market transactions [3][5]. Given the results of forum research in contributing to our understanding of cybercriminals, it appears pertinent to also scrutinize hacker IRC channels. Thus, it is worthwhile to explore past research on IRC channels outside of the cybercriminal context in order to gain insights for conducting our own study.

B. IRC-based Research

With the advent of web 2.0, researchers have become interested in closely examining the behaviors of individuals in virtual settings. Many studies in this research stream have focused investigation on virtual community usage and user participation behaviors within the contexts of web forums, blogs, and even IRC channels. Such research has yielded many interesting findings relevant to explaining various facets of virtual community participation. Insights and perspectives can be borrowed from previous works utilizing IRC data to help operationalize our study.

In particular, IRC channels exist inside their own protocol and require special consideration for collection. Instead of using an automated crawler to collect data, such as in the case of web forums, IRC channel data must be collected in real time [10]. In past research, automated chat-logging bots have been adopted to the IRC protocol and utilized for collection [6]. In essence, such bots simply connected to various IRC channel and passively listened for data to collect.

Several strategies can be taken when deploying bots to ensure comprehensive collection of IRC data [6]. First, several chat logging bots can be deployed to a single channel. In the event that a bot is removed from channel or otherwise disconnected, collection will continue as other bots are connected. Second, connected bots can be swapped out with replacements to avoid remaining idle in a single IRC server for too long. This behavior may help reduce the chances of a bot being removed or banned for idling. Lastly, bots can be designed as intelligent agents that can interact with other channel participants. For example, a bot may be programmed to detect a particular topic of interest being discussed by channel participants, and can be designed to react with pre-scripted messages to avoid idling or suspicions of bot activity.

However, various challenges can be encountered that may inhibit the effectiveness of automated chat-logging. Primarily, when dealing with hacker communities, identity obfuscation is worthwhile as an additional security measure for researchers. Without anonymizing bot network traffic, participants of the hacker community may be able to identify the source of bots and retaliate. Additionally, despite the aforementioned collection strategies, IRC bots may still be forcibly disconnected from channels if bot-activity is suspected by channel operators. Special consideration must be paid to ensure no chat data is lost in these instances, and circumvention of access restrictions must be built into our collection system.

III. SYSTEM DESIGN

An IRC chat data collection system must fulfill several requirements. First, the system must be automated and scalable to collect data from multiple IRC channels simultaneously. Ideally, researchers would only supervise the system's activity to reduce usage effort. Second, the system must ensure comprehensive collection. Several bot-deployment strategies can be taken to decrease the likelihood of missing chat data. Lastly, the system should be resilient and capable of circumventing access restrictions of hacker IRC communities. If bot-like activity is suspected, the system may be barred from connecting to a particular IRC channel; through practice of identity obfuscation and anonymization of network traffic, such restrictions can be circumvented. A high-level design of our system can be seen in Figure 1.

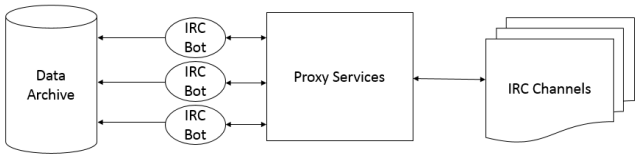

Figure 1 – IRC Collection System Design

The system's main responsibility is to maintain a log for all the messages collected in the channels. The log contains information such as, the time, the channel name, the user name and the detail message. Any hyperlinks shared by channel participants are stored in a separate file that can be later fed into a web crawler for additional analysis of hacker contents.

Based on researcher input within a text configuration file, the system creates multiple instances of bots that automatically connect to a designated IRC server and channel. Additionally, a single bot instance can be configured to connect to several channels simultaneously, allowing for the collection of more data while using less CPU resources. Instanced bots can also be configured to automatically attempt rejoining channels when disconnected to avoid data loss.

To improve the robustness of our system, two or more bots are deployed to a single channel. If any single bot should require reconnection to the target channel, it assumes a new handle in attempt to avoid triggering suspicions of other channel participants. Thus, the bots have the capability to assign random name to itself upon join.

Over the time we noticed that passive IRC users, i.e. those who do not show much activity in the channel are frequently removed from the channel by the channel operator. This kind of activity helps the channel free up resources by disconnecting idle users and potential chat logging bots. To counter this and to maintain robustness, our bots automate some activity such as automatically reconnecting on their own at random intervals or by probing the server for channel information to avoid being removed for idling

Our IRC bots also have the capability to retrieve information about the users directly. A bot sends an IRC 'whois' request to retrieve information about a particular user. The information retrieved contains data such as the user's IP address, IRC operator privileges, channels joined etc. By viewing other channels a particular user is connected to, we are able to potentially identify and collect more hacker IRC channels. Therefore, the collection system periodically request the "list of joined channels" from each participant in the connected channel. Popular channels that multiple users are participating then can be monitored by the IRC system by deploying additional bots or by simply joining the channel with existing bots. The data from the new channel is then logged and stored in our archive.

We also expand our observation scope by identifying more channels using keyword matching based on /list command in IRC. A separate bot issues an IRC list command which gives the list of the entire channel with their description created at a particular server. Then using string matching for keywords like "hack", "hackers", "anonymous" etc we can get channels of interest in no time which are then used by other bots for logging.

All bot network traffic is routed through proxy servers or the Tor peer-to-peer anonymization network. This helps anonymize the source of bots, helping protect researcher identity and security. Additionally, some channels may exist as deep web hidden services that are only accessible through peer-to-peer connections on the Tor network. These channels can also be logged by our system.

IV. Preliminary Results

To test our implementation of a hacker IRC collection system, we identified four IRC channels suitable for collection. Similar to previous hacker community research, we utilize keyword searches to identify potential hacker IRC channels, e.g. "Carding chat server" and "blackhat irc". After identification, we deployed automated IRC chat logging bots were deployed to identified IRC channels. Multiple bots issued from various hosts to ensure collection and to avoid gaps in data from dropped connections, bans, etc. We also practiced identity obfuscation by routing Internet traffic generated by our collectors through the Tor peer-to-peer anonymization network. We observed captured data and selected our most popular IRC channels for this research, as seen in Table 1.

Server	Channel	# of Users	# of Messages	Collection Date Range
irc.anonops.com	#anonops	1,645	97,587	11/20/13 – 3/2/14
Irc.hack5.org	#hak5	209	12,362	2/13/14 – 3/2/14
cbadanhgoo6oamul.onion	#salt	215	14,658	2/9/2014 – 3/2/14
6dvj6v5imhny3anf.onion	#freeanons	161	11,532	2/9/2014 – 3/2/14

Table 1 – IRC collection status. In the #cc-trade channel, we discovered that the 50 users were in fact bots that would constantly spam black market and carding related advertisements.

The #anonops community is particularly interesting due to its relevance to the *Anonymous* hacking community. *Anonymous* has been commonly referred to as a collective of hacktivists that regularly disrupt web-enabled resources for a variety of societal or political causes. *Anonymous* makes use of IRC, with the #anonops community being the particular target of a prior government investigation [2]. We were also able to identify two ".onion" deep web hidden services. These IRC channels are only accessible when connected to the Tor peer-to-peer anonymization network, providing participants with an additional layer of anonymity and security.

We first identified the #anonops channel, and later discovered the other channels in our collection. The IRC collection system we developed was able to successfully remain connected to the IRC channels to collect data. We observed that the collection system was able to mitigate potential stoppages in data collection due to disconnection.

V. Future Work

We plan to develop a honeypot environment for additional collection of hacker IRC data. Honeypots are systems that are configured to simulate computer environments with software vulnerabilities; the idea is to have wild malware exploit honeypot vulnerabilities so that the malware can be captured and studied in a sandboxed environment. All code execution, system changes, and network traffic are tracked and logged within a honeypot, letting security researchers understand the nature of some particular malware [11]. Honeypot systems managed by researchers can help provide data for malware traffic and behavior analysis [12]. Honeypot approaches towards collection can help identify botnet command and control channels by observing outbound network connections attempted by captured malware.

Figure 2 represents an overview of the proposed environment, which consists of the following components. IRC Server will support the interaction with the rest of the IRC network and also log all IRC messages. Autonomic Bot Generator will be responsible for generating bots that provide interaction mechanism with the environment. The bot behaviors, types, and number are enforced based on a preset policy. Autonomic Monitoring is responsible for picking up all the IRC packets, and it will have Network policies that define which ports to monitor and when. IRC Message Extraction will extract IRC messages from IRC packets, and categorize them into different IRC message types. File Extraction will detect file transfer and extract files from communications; Conversation Historian will build conversations from the IRC messages and storing those for further analysis. Feature Extraction and Reduction will extract all the features needed to perform the analysis from the IRC messages. Malware Analysis and Social Media Analyzer are the analytical engines for the environment; one focusing on detecting malware and the other on detecting, classifying, measuring, and tracking the formation, development, and spread of topics, ideas, and concepts in cyber attacker social media communication.

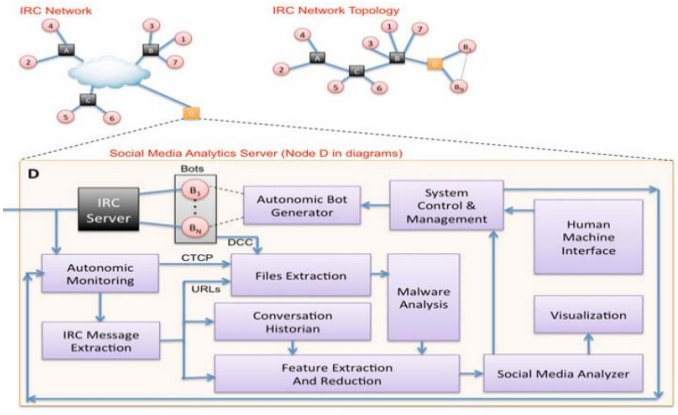

Figure 2 – Future Work on IRC Collection System

VI. Conclusion

This paper outlines a preliminary hacker IRC collection system, including hidden services such as Tor .onion links. In future work, we plan to expand the scope of our system by deploying honeypots to identify potential botnet command and control channels. Several other additional features are planned to help support other forms of research, such as hacker social media analysis (through our Conversation Historian) and by malware analysis. The collection system can automatically identify and provide data to support research on advancing currently cybersecurity capabilities.

Acknowledgements

This work was supported by the National Science Foundation under Grant No. No. SES-1314631 and also under Grant No. DUE-1303362.

References

[1] National Science and Technology Council (2011). *Trustworthy Cyberspace: Strategic Plan for the Federal Cybersecurity Research and Development Program* (pp. 1–19).

[2] Schone, M., Esposito, R., Cole, M., & Greenwald, G. (2014). *War on Anonymous: British Spies Attacked Hackers.* National Broadcasting Company (NBC). http://www.nbcnews.com/news/investigations/war-anonymous-british-spies-attacked-hackers-snowden-docs-show-n21361

[3] Motoyama, M., McCoy, D., Levchenko, K., Savage, S., & Voelker, G. M. (2011). An analysis of underground forums. *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference - IMC '11*, 71.

[4] Benjamin, V. A., & Chen, H. (2013). Machine Learning for Attack Vector Identification in Malicious Source Code. *IEEE Intelligence and Security Informatics*, 21–23.

[5] Radianti, J. (2010). A Study of a Social Behavior inside the Online Black Markets. *2010 Fourth International Conference on Emerging Security Information, Systems and Technologies*, 88–92.

[6] Fallmann, H., Wondracek, G., & Platzer, C. (2010). Covertly Probing Underground Economy Marketplaces. *Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, 101–110.

[7] Holt, T. J., & Lampke, E. (2010). Exploring stolen data markets online: products and market forces. *Criminal Justice Studies: A Critical Journal of Crime, Law, and Society*, *23*(1), 33–50.

[8] Benjamin, V., & Chen, H. (2012). Securing Cyberspace : Identifying Key Actors in Hacker Communities. *IEEE Intelligence and Security Informatics*, 24–29.

[9] Holt, T. J., Strumsky, D., Smirnova, O., & Kilger, M. (2012). Examining the Social Networks of Malware Writers and Hackers. *International Journal of Cyber Criminology*, *6*(1), 891–903.

[10] Sinha, T., & Rajasingh, I. (2014). Investigating substructures in goal oriented online communities: Case study of Ubuntu IRC. *2014 IEEE International Advance Computing Conference (IACC)*, 916–922.

[11] II, C. J. M., & Chen, H. (2008). Botnets, and the CyberCriminal Underground. *IEEE International Conference on Intelligence and Security Informatics 2008*, 206–211.

[12] Holt, T. J., & Kilger, M. (2012). Know Your Enemy : The Social Dynamics of Hacking. *The Honeynet Project*, 1–17.