Universal Peer-to-Peer Network Investigation Framework

Mark Scanlon and M-Tahar Kechadi
School of Computer Science and Informatics,
University College Dublin,
Belfield, Dublin 4, Ireland.
Email: {*mark.scanlon,tahar.kechadi*}*@ucd.ie*

***Abstract*—Peer-to-Peer (P2P) networking has fast become a useful technological advancement for a vast range of cybercriminal activities. Cybercrimes from copyright infringement and spamming, to serious, high financial impact crimes, such as fraud, distributed denial of service attacks (DDoS) and phishing can all be aided by applications and systems based on the technology. The requirement for investigating P2P based systems is not limited to the more well known cybercrimes listed above, as many more legitimate P2P based applications may also be pertinent to a digital forensic investigation, e.g, VoIP and instant messaging communications, etc. Investigating these networks has become increasingly difficult due to the broad range of network topologies and the ever increasing and evolving range of P2P based applications. This paper introduces the Universal Peer-to-Peer Network Investigation Framework (UP2PNIF); a framework which enables significantly faster and less labour intensive investigation of newly discovered P2P networks through the exploitation of the commonalities in network functionality. In combination with a reference database of known network protocols and characteristics, it is envisioned that any known P2P network can be instantly investigated using the framework. The framework can intelligently determine the best methodology dependant on the focus of the investigation resulting in a significantly expedited evidence gathering process.**

***Keywords*-Peer-to-Peer; P2P; Botnet; Mitigation; Computer Forensics; Cybercrime; Investigation**

I. Introduction

P2P networks are widely used as a low-overhead, efficient, self-maintaining, distributed alternative to the traditional client/server model across a broad range of areas. As a result of these desirable attributes, the technology also lends itself well to being utilised for malicious purposes due to the minimal setup and maintenance costs involved. The financial impact of malicious P2P networks can be significant. In 2008, the Motion Picture Association of America reported that Internet piracy cost the film industry $7 billion USD, with the majority of that facilitated by P2P file-sharing networks [1]. In 2012, the Zeus botnet is estimated to have caused damages of over $100 million USD since its discovery in 2007 [2].

In 1999, Napster catapulted the relatively new concept of P2P Internet file-sharing into the mainstream [3]. It facilitated regular home Internet users in the sharing of their digital music collections with millions of other Napster peers irrespective of who they were or their geolocation. The ease of use, vast library of available content, perceived anonymity and zero cost model facilitated Napster to grow rapidly. Its rise in popularity also coincided with the release and popularity of portable MP3 players [4]. Napster was subsequently sued by the Recording Industry Association of America (RIAA) and was ordered to shut down by the US court in 2001. Due to its centralised topology, the service was easily terminated, i.e., the "register" of connected nodes and associated shared content was stored on Napster's servers and without this register the system could not operate. A decentralised system is much more difficult to disrupt. Taking down a single node or a subset of nodes has minimal impact on the network as a whole [5].

A. P2P Networks

Since P2P networking has become mainstream, the technology has been deployed across a broad range of systems and services. While the level of variation in topologies is significant, all P2P networks must share a number of common attributes:

1) Capability to connect to the network (bootstrapping) – When a new node wishes to join the network, it must have the ability to contact at least one other active participant in the network. Depending on the network design, this may take the form of a hardcoded list of active nodes (typical in a decentralised topology) or a list of bootstrapping servers (typical in a centralised topology) [6].
2) Record Maintenance of Active Nodes – In a decentralised network the peers themselves must all contribute to the recording of active nodes on the network. No single peer has the entire list, with each peer contributing to a collective distributed database, typically a distributed hash table (DHT). In a centralised design, this duty falls on the controlling server(s). As each new node comes online, it announces its presence to the database maintainer and requests a list of other active peers to begin working.
3) Query/Order/File Propagation – In order for a P2P network to fulfil whatever the purpose it was designed for, intra-peer communication is requisite. As a result

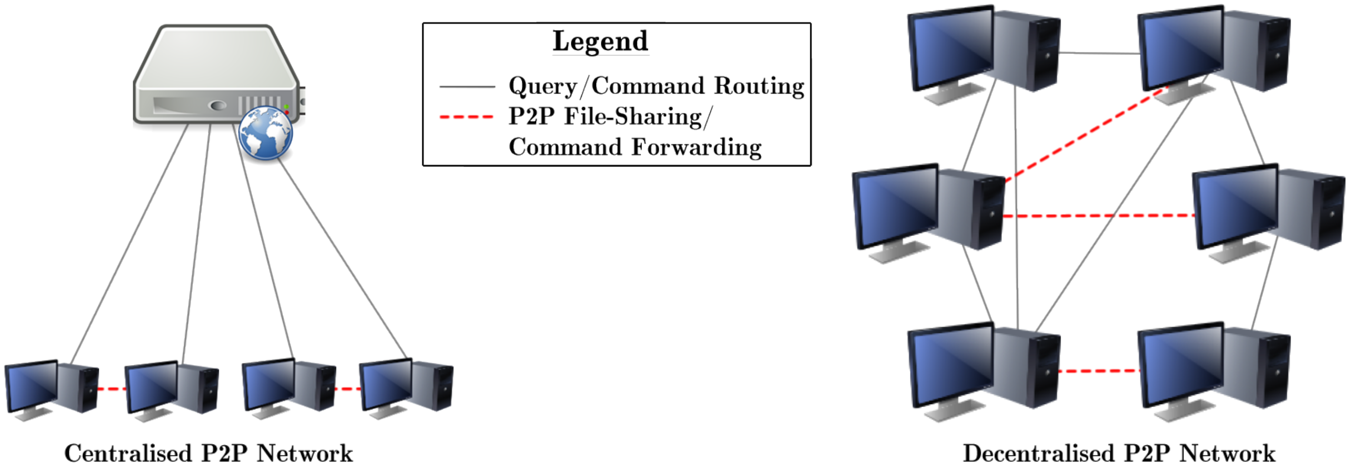
Legend
Query/Command Routing
P2P File-Sharing/ Command Forwarding
Centralised P2P Network
Decentralised P2P Network

Figure 1. A Comparison of Centralised (left) and Decentralised (right) P2P Network Architectures.

of this necessity, each peer must be able to receive requests or commands and pass these communications onto other known peers.

4) Software Maintenance – The P2P enabled binary can quickly become outdated. The upgrade process must be simple to perform while maintaining node uptime. While newer versions of the application might have additional functionality, it must to ensure backwards compatibility otherwise the network as a whole may suffer.

In this paper, we introduce a framework which enables forensic investigators and researchers to fast-track the investigation of any P2P network. The framework exploits many of the common attributes of these networks outlined above. As can be seen in Fig. 1, each node on a P2P network has two main functions. The first involves participating in the maintenance of the network itself – it is aware of a number of other nodes, and is in communication with a subset of the overall population. From the analysis of the P2P communication of an active node in the network, common communications can be identified, e.g., peer discovery, query/command/file propagation, etc. Once each of these communication patterns are identified, the behaviour of an active node can be recorded in a centralised shared database. The sharing of identified patterns will aid in the elimination of duplicated work by forensic investigators. As the usage of UP2PNIF increases, the database of identifiable traffic patterns and the value of using the framework will become significantly greater.

II. P2P Network Investigation Types

Once the communication method of the undocumented network is reverse engineered through traditional means, crawling the network using UP2PNIF will then be possible. UP2PNIF is designed to aid in the following investigation types:

A. Evidence Collection

In order to combat the unauthorised downloading of copyrighted material, many countries have implemented a three to six strikes “graduated response” system whereby repeat offenders will have their Internet service discontinued for a defined penalty period [1]. In order for such systems to operate, evidence must be gathered to prove that infringement has taken place. In a botnet investigation, the evidence collected might take the form of the commands issued, the origin of these commands, the targets of an attack, etc. The concept of evidence collection is easily understood with respect to illegal file-sharing and botnet investigations, but it can equally apply to more legitimate P2P applications such as VoIP, e.g., Skype [7], or instant messaging services, e.g., AIM or MSN Messenger [8].

B. Anatomy

Investigating the anatomy of a particular network involved the analysis of the client binary’s behaviour and analysis of the network communication patterns. This type of investigation attempts to classify the system as centralised/decentralised, Client-Server/P2P hybrid or solely P2P based command and control. An anatomy documentation investigation can continue past the network architecture of the system to cover some of the counter-detection and anti-forensic techniques employed. For example, Goel et al. discovered that “Agobot” had a built in defence mechanism to terminate the execution of a remotely upgradable list of over 610 anti-virus programs [9]. The Storm worm was also engineered to aggressively use the distributed botnet to collectively attack anyone who attempted to reverse engineer it [10].

C. Wide-Area Measurement

This concentrates on attempting to enumerate the population of the network, its bandwidth, its computational power

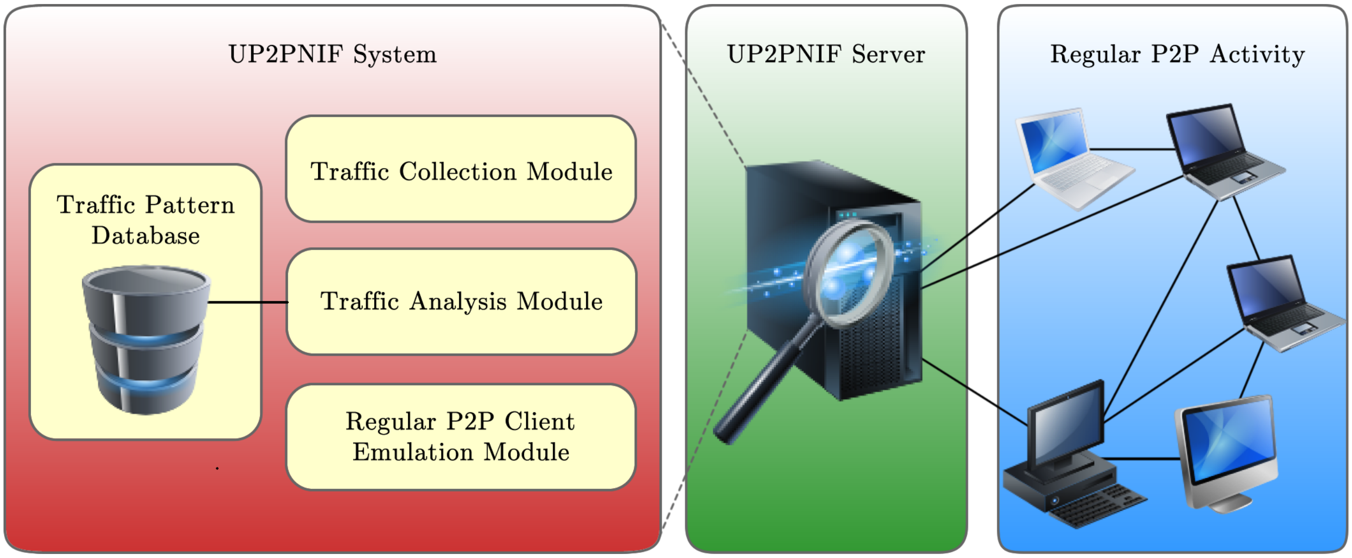

Figure 2. UP2PNIF architecture with the regular P2P activity on the right and the UP2PNIF server on the left.

or its often multi-faceted goals. Gathering the population of a P2P network is traditionally a non-trivial task as the number of nodes ever connecting to any single node or to a Command and Control (C&C) server may only count for a small proportion of the participating nodes. There are two definitions of a P2P networks size, as specified by Rejab et al. [11]:

1) Footprint – This indicates the aggregated total number of machines that have been compromised over time.
2) Live Population – This measure denotes the number of compromised machines that are concurrently in communication with each other.

In the case of measuring a documented network, e.g., BitTorrent, a custom crawler must be built which can efficiently collect peer information. In the case of an undocumented network (e.g. a P2P botnet), a relatively straightforward method for measuring the population of the network is to run a bot on a deliberately infected machine and monitor the resultant network traffic. The number of IP addresses the infected node is in communication with can be easily counted having eliminated all non-botnet related network traffic. It is unsafe to assume that a single node will ultimately communicate with every other node in the network over any reasonable time frame. Traditional enumeration investigation increase the number of infected machines (physically or virtually) and amalgamate the results should lead to a more accurate representation. Using UP2PNIF, regular client usage can be amplified to achieve the required results in a significantly more timely manner.

D. Takeover

Botnet takeover involves a third party gaining control of a botnet from its owner. This third party could be law enforcement, researchers or another cybercriminal. Once control has been gotten of the botnet, the new botmaster is able to issue commands, update configurations and operate the botnet as desired. In 2009, Stone-Gross et al. successfully took over the Torpig botnet for 10 days [12]. During this time, the researchers identified more than 180,000 compromised machines and were sent over 70GB of automatically harvested personal information. The number of discovered unique Torpig bot IDs and corresponding number of IP addresses was observed to be 182,914 and 1,247,642 respectively [13]. The discrepancy between the number of bots and IP addresses found is accountable by network effects such as DHCP churn and NAT, as described in greater details in Section V below.

III. UP2PNIF

A. System Architecture

The software architecture of the system consists of four main components, as can be seen in Fig. 2. The framework is designed to operate in either a forensic laboratory, in a cloud environment or in a remote, portable “on-the-fly” scenario. When the system is operating on a P2P network, it appears to each regular node as another regular client on the network. Depending on the investigation type, traffic can be collected from a P2P client running on the same machine as the tool or it can act as an intermediary proxy between a P2P node and the rest of the network.

In the on-the-fly usage scenario, the system’s task is solely one of live network evidence collection. This evidence will be collected and stored either on a remote hard drive or uploaded directly to cloud storage assuming network bandwidth capabilities. Regular time-stamping and hashing will take place to ensure forensic integrity, as outlined in more detail below. Once this data is collected, it is immediately available for analysis to cluster and classify the traffic.

Byung et al. proposed in 2009 a methodology for improving botnet size estimates through the implementation of a botnet crawler, called Passive P2P Monitor (PPM) [14]. PPM acts as though it were the same as any other node on the network by implementing the "Overnet Protocol" of the Storm botnet. This method involves mimicking the functionality of a regular bot, contributing to the Distributed Hash Table, forwarding commands, etc. For each peer the crawler connects to, it has the ability to exchange peer information. In this manner, a list of all known peers on the network can be compiled by sequentially exchanging peer information with all newly discovered peers.

B. Traffic Collection Module

This module monitors the network traffic of a specific machine, or a group of machines. The packet sniffing is conducted using "libpcap" (or its windows alternative, "winpcap") [15]. This module is also responsible for packaging the collected data streams and associated metadata into a digital evidence bag and, if necessary, securely transferring the evidence to an external storage device or system.

C. Traffic Pattern Database

This component stores the patterns of known networks. The types of metadata stored for each network include common hostnames and IP addresses, peer discovery methods and frequency of updates, common commands, update methods, etc. This database is used for reference by the analysis module to aid in the identification of newly gathered network traffic.

D. Traffic Analysis Module

The module analyses the collected packets. The frequency, content/pattern and destination of the packets contribute to the identification of the traffic. In order for a P2P network to function, each peer must regularly check-in with a centralised server or with other active peers at specific intervals. Each suspected packet can be compared to the above database of known network usage patterns. This is particularly useful in the identification of botnets as each specific botnet system can be used by numerous botmasters in the creation of many separate networks.

E. Client Emulation Module

A client application is capable of performing a number of differing forensic investigations. Depending on the specific network, it may be possible to conduct each of the following investigations as required by the case at hand:

1) Network Enumeration – This investigation concentrates on attempting to enumerate the population of the entire network, as well as the combined bandwidth and computational power. Gathering the population of a network is traditionally a non-trivial task, as the number of nodes simultaneously connecting to any one node generally only accounts for a small subsection of the entire network. The client emulation module overcomes this limitation by amplifying regular client usage. Care is taken to ensure that any single node is not communicated with in a "suspicious" manner, ensuring consistency with regular P2P traffic patterns.
2) Network Usage – This investigation is focused on finding out what the network is being used for. For example, in the case of a P2P botnet, the investigation might be targeted at finding out what commands each node receives, i.e., what is the botnet being used for? In this scenario, the commands/file distributed through the network are recorded for analysis. Due to the framework partaking in the network as though a regular node, should encryption be employed, the framework can employ the same encryption standards as a regular node.
3) Network Anatomy/Modelling – This investigation is focused on attempting to understand the design and structure of the network and client software. Based on the gathered evidence, the network topology can be extrapolated. The results obtained can aid in determining whether the network is centralised, decentralised or a hybrid, the frequency of intra-peer communication, the contribution of each node to the maintenance of a DHT, the attack vector utilised by the malware, etc.

F. Forensic Integrity

Due to the sensitive nature of digital evidence collection, it is imperative that the data collected by any forensic tool is absolutely verifiable and identical to the original source. This integrity is ensured in the UP2PNIF system through the implementation of a new live digital evidence bag. This evidence bag will record all relevant information, e.g., IP addresses, packets, running processes, etc. Once any network traffic is collected, each packet is time-stamped and logged. The time-stamping facilitates real-time event reconstruction packet by packet, emulating the original traffic.

The integrity is insured in the UP2PNIF system through the implementation of regular hash checking on the data being collected using SHA-512, a 512-bit secure hashing algorithm. The system collects a stream of information, the stream itself is hashed and both are stored on the external drive or can be uploaded to secure cloud storage. During the transmission process, the integrity of each of the chunks being transferred is maintained due to a SHA-512 hash being computed as the chunk is being transmitted. Server-side, once the transmission is completed, a SHA-512 hash is taken on the chunk and verified against the original. If these hashes do not match, i.e., the integrity of that chunk has been compromised in transmission, a failure notification is sent to the client, which queues that chunk up again for transmission.

G. Comparison with Existing Tools

Many of the existing P2P network investigation tools are built focusing on a single network. For almost as long as P2P networks have been in existence, there have been eavesdropping and crawling software built to investigate them [16]. In order to centralise the collective intelligence of forensic researchers, storing the reverse engineered network protocols and behaviours in a centralised database will be greatly beneficial for all stakeholders concerned with P2P investigation.

IV. Advantages

1) Compatibility – One advantage of using the UP2PNIF system is that irrespective of what design or configuration is used by the network, an investigation on the network should be able to commence as quickly as possible. The framework is capable of aiding in the identification and fast-tracked emulation of any P2P network.
2) Cost – The cost involved in running the UP2PNIF system is minimal; mainly the costs associated with the server costs used for investigation. The system would ideally run on a high-end server with a high-speed Internet connection. It would also be necessary to have access to a large amount of storage, be it local storage, a connected NAS (network attached storage) or secure online storage. Running the entire system in the cloud would offer significant advantages over a physical setup as the speed and number of concurrently running investigations will be greatly improved. It is also advantageous for the investigation of a physical machine to remotely store the collected evidence in the cloud, or to send it back to the forensic laboratory.
3) Automated Identification – This feature of the UP2PNIF system results in users requiring little technological network knowledge to operate. Due to the centralised database of known network traffic patterns, it will ultimately result in forensic investigation being possible in more places at once, e.g., in a law enforcement scenario, each police station would potentially have the ability to identify and collect P2P evidence without the requirement for an on-site digital forensic specialist. The contribution of network investigation specialists towards the centralised database will ensure that newly discovered networks will be added as soon as possible.
4) Speed – While each individual network detection and identification can take some time, once the network has been reverse engineered the investigation can begin almost instantaneously with the ability for multiple investigations to take place simultaneously. Should a sufficiently complete identification database be contributed to over time, the ease of identifying variants of the same system would be greatly reduced over a manual, single investigator approach.

V. Potential Limitations

There are a number of obstacles to the investigation of P2P networks. Most of the obstacles outlined below are applicable to both documented and undocumented networks [18]:

1) Dynamic Host Configuration Protocol (DHCP) - Due to a typical lease from an Internet service provider lasting in the order of 1-7 days, dynamic reallocation of the same IP address may result in two or more peers participating in the network appearing as a single peer impacting upon the accuracy of an enumeration focused investigation.
2) Proxy servers - Similar to the issue caused by DHCP, any nodes that access the Internet through the same transparent or anonymous proxy server will also appear as a single node to the outside world.
3) Identification of peers using anonymous Internet services – This facilitates the identification of services such as Tor (The Onion Router) and I2P (Invisible Internet Project). By comparing the IP addresses discovered during the investigation with a list of known Internet traffic proxy or pass-through services, such as that maintained by MaxMind Inc. [19], the quality of the results collected can be greatly improved.
4) Network Address Translation – Numerous machines behind a shared router may appear to the outside world as a single machine as they share a single IP address.
5) Encrypted Communication - Should the network employ encrypted communication, the only method available for investigation is to attempt to reverse engineer the client software. The decryption key for any incoming commands or peer discovery must be stored within the client. Once the encryption specification is discovered, the modular framework can employ the same encryption methods.
6) Difficulty in Take Down – In order to take down a P2P network, it is often a matter of discovering their weak spot. Traditionally this has meant attempting to take down their centralized server [20]. However, with the popularity of employing a fully decentralized network design, the ability to take down such a network has been made considerably more difficult. Should the client be reverse engineered, it is possible that the network could be disturbed or even imploded, e.g., through the issuing of an uninstall command to each infected node in a botnet.

The limitations outlined above with respect to detecting unique peers as a result of potentially several peers appearing

Table I
BITTORRENT BEHAVIOUR PROFILE

Feature	Value
fileSharing	true
botnet	false
centralised	true
decentralised	true
configFile	true
encrypted	false
httpPeerDisc	true
httpFreq	1800
peerExchange	true
peerExchangeFreq	600
dht	true
dhtFreq	1250

Table II
BITTORRENT NETWORK COMMUNICATION FORMAT

Feature	Value
httpFormat	%httpURL%?info_hash=%fileID%&peer_id=%peerID% &port=6881&numwant=200&compact=1&uploaded=0 &downloaded=0&left=0
pexFormat	%peerIP%:%peerPort%
dhtFormat	%dhtIP%?id=%btID%&info_hash=%fileID%

under a single IP address can be circumvented on networks that employ a unique ID number, such as the Torpig botnet [13], the BitTorrent DHT [18], etc. Other heuristic metadata, such as client version information, detected data speeds/latency and the list of available files (in the case of file-sharing networks) can each contribute to unique identification. Participating in the network as a regular client means that the network must be fully understood or reverse engineered, including the encryption methods. By being a part of the network and appearing as any other node, any issues traditionally involved in attempting to decrypt captured network packets are rendered irrelevant.

VI. PROOF OF CONCEPT

In order to prove the viability of the framework outlined above, a proof of concept was built and tested using the P2P file-sharing network, BitTorrent. This network was selected due to the popularity of the network and due to the variation of differing methods for communication and peer discovery. There are a number of methods that a BitTorrent client can attempt to discover new peers who are in the swarm:

1) Tracker Communication – BitTorrent trackers maintain a list of seeders and leechers for each BitTorrent swarm they are currently tracking. Each BitTorrent client will contact the tracker intermittently throughout the down- load of a particular piece of content to report that they are still alive on the network and to download a short list of new peers on the network.
   Tracker communication is similar to a new bot infection on a machine requested a list of peers from a centralised command & control server, such as that used by Nugache [20], .
2) Peer Exchange (PEX) – Peer Exchange is a BitTorrent Enhancement Proposal (BEP) whereby when two peers are communicating, a subset of their respective peer lists are shared during the communication.
3) Distributed Hash Tables (DHT) – Within the confounds of the standard BitTorrent specification, there is no intercommunication between peers of different BitTorrent swarms. Azureus/Vuze and uTorrent contain mutually exclusive implementations of distributed hash tables as part of the standard client features. These DHTs maintain a list of each active peer using the corresponding clients and enables cross-swarm communication between peers. Each peer in the DHT is associated with the swarm(s) in which he is currently an active participant.
4) Local Peer Discovery – This extension to the BitTorrent protocol enables the discovery and exchange of data with peers on the same local area network (LAN) as the client software. Its purpose is to alleviate the volume of traffic routed through while taking advantage of the often greater LAN data transmission speeds. In practice, this speed advantage is only beneficial if two or more peers on the same LAN are participating in the same swarm.

To start, the basic profile for BitTorrent was recorded in the network database, as can be seen in Table I. Alongside this identifying information, the format of the communication methods are also stored, as can be seen in Table II. A hierarchical object oriented approach to network investigation was implemented for each of the BitTorrent peer communication methods. In testing the crawler, it was found that by employing a hierarchical approach facilitated much of the job organisation and data processing.

VII. CONCLUSION AND FUTURE WORK

P2P networks are a desirable choice for the execution of a number of cybercrimes as they afford the perpetrators minimal investment to conduct their crimes. The ideal design for a P2P network from a counter-forensic standpoint is one that is completely decentralised, utilises unique encryption methods and operates on a bespoke network protocol for communication. Investigation of such a network may prove particularly difficult using traditional methods. However, the utilisation of a system such as UP2PNIF will greatly improve the time from initial detection of a new network, to conducting the required investigation resulting in prompt, more efficient response and evidence gathering.

This system is capable of exploiting the fundamental requirement for any new node to have some starting point in seeking out other active nodes on the network. This will always leave a vulnerability for traffic detection, monitoring and emulation. The fact that many P2P networks are moving towards entirely decentralised designs is counter intuitively an advantage for network investigation due to the added difficulty in distributed investigation detection. In order for UP2PNIF to reach its potential, it requires the use of the framework and methodology with a comprehensive variety of P2P networks. The greater number of networks it identifies, the larger the database of known patterns will become and the performance of the overall system will greatly improve.

While the framework outlined above is capable of detecting, identifying and partaking in a P2P network, there remains a significant legal hurdle. Due to the cross-border nature of Internet systems, international law needs to clearly provision for the investigation of these networks [21]. There are also significant ethical considerations in the investigation of malicious P2P systems. Should the forensic investigator implode a detected botnet ("freeing" the infected machines) or disrupt a DDoS attack? Has he the right?

Acknowledgement

This work has been co-funded by the Irish Research Council and Intel Ireland Ltd. through the Enterprise Partnership Scheme. The authors also wish to acknowledge the contribution of Amazon for supporting this research through an Amazon Web Services research grant.

References

[1] D. Serbin, "The graduated response: Digital guillotine or a reasonable plan for combating online piracy?" *Intellectual Property Brief*, vol. 3, no. 3, p. 4, 2012.

[2] M. Riccardi, R. D. Pietro, M. Palanques, and J. A. Vila, "Titan's revenge: detecting zeus via its own flaws," *Computer Networks*, 2012.

[3] M. Giesler and M. Pohlmann, "The anthropology of file sharing: Consuming napster as a gift," *Advances in consumer research*, vol. 30, pp. 273–279, 2003.

[4] E. Van Buskirk, "Introducing the world's first mp3 player," *MP3 Insider*, vol. 27, 2005.

[5] R. Jaiswal and S. Bajgude, "Botnet technology," in *3rd International Conference on Emerging Trends in Computer and Image Processing (ICETCIP'2013)*, January 2013, pp. 169–175.

[6] M. Conrad and H.-J. Hof, "A generic, self-organizing, and distributed bootstrap service for peer-to-peer networks," *Self-Organizing Systems*, pp. 59–72, 2007.

[7] S. A. Baset and H. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *IEEE infocom*, vol. 6, 2006, pp. 23–29.

[8] R. B. Jennings, E. M. Nahum, D. P. Olshefski, D. Saha, Z.-Y. Shae, and C. Waters, "A study of internet instant messaging and chat protocols," *Network, IEEE*, vol. 20, no. 4, pp. 16–21, 2006.

[9] S. Goel, A. Baykal, and D. Pon, "Botnets: the anatomy of a case," *Journal of Information Systems Security*, 2006.

[10] N. M. Mukamurenzi, "Storm worm: A p2p botnet," Master's thesis, Department of Telematics, Norwegian University of Science and Technology, 2008.

[11] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging," in *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, ser. HotBots'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 5–5. [Online]. Available: http://dl.acm.org/citation.cfm?id=1323128.1323133

[12] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009, pp. 635–647.

[13] B. Stone-Gross, M. Cova, B. Gilbert, R. Kemmerer, C. Kruegel, and G. Vigna, "Analysis of a botnet takeover," *Security & Privacy, IEEE*, vol. 9, no. 1, pp. 64–72, 2011.

[14] B. Kang, E. Chan-Tin, C. Lee, J. Tyra, H. Kang, C. Nunnery, Z. Wadler, G. Sinclair, N. Hopper, D. Dagon *et al.*, "Towards complete node enumeration in a peer-to-peer botnet," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*. ACM, 2009, pp. 23–34.

[15] S. McCanne, C. Leres, and V. Jacobson, "Libpcap," June 2012.

[16] S. Saroiu, P. K. Gummadi, and S. D. Gribble, "Measurement study of peer-to-peer file sharing systems," in *Electronic Imaging 2002*. International Society for Optics and Photonics, 2001, pp. 156–170.

[17] P. Wang, L. Wu, B. Aslam, and C. C. Zou, "A systematic study on peer-to-peer botnets," in *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*. IEEE, 2009, pp. 1–8.

[18] M. Scanlon, A. Hannaway, and M.-T. Kechadi, "A week in the life of the most popular bittorrent swarms," *5th Annual Symposium on Information Assurance (ASIA'10)*, 2010.

[19] Maxmind Inc. (2013, Apr.) Geolite country database. [Online]. Available: http://www.maxmind.com

[20] R. Schoof and R. Koning, "Detecting peer-to-peer botnets," System and Network Engineering, University of Amsterdam, Tech. Rep., February 2007.

[21] D. Dittrich, F. Leder, and T. Werner, "A case study in ethical decision making regarding remote mitigation of botnets," *Financial Cryptography and Data Security*, pp. 216–230, 2010.