TorWard: Discovery of Malicious Traffic over Tor

Zhenyu Ling*†, Junzhou Luo*, Kui Wu†, Wei Yu‡ and Xinwen Fu§
*Southeast University, Email: {zhenyuling, jluo}@seu.edu.cn
†University of Victoria, Email: wkui@cs.uvic.ca
‡Towson University, Email: wyu@towson.edu
‡University of Massachusetts Lowell, Email: xinwenfu@cs.uml.edu

***Abstract*—Tor is a popular low-latency anonymous communication system. However, it is currently abused in various ways. Tor exit routers are frequently troubled by administrative and legal complaints. To gain an insight into such abuse, we design and implement a novel system, *TorWard*, for the discovery and systematic study of malicious traffic over Tor. The system can avoid legal and administrative complaints and allows the investigation to be performed in a sensitive environment such as a university campus. An IDS (Intrusion Detection System) is used to discover and classify malicious traffic. We performed comprehensive analysis and extensive real-world experiments to validate the feasibility and effectiveness of *TorWard*. Our data shows that around 10% Tor traffic can trigger IDS alerts. Malicious traffic includes P2P traffic, malware traffic (e.g., botnet traffic), DoS (Denial-of-Service) attack traffic, spam, and others. Around 200 known malware have been identified. To the best of our knowledge, we are the first to perform malicious traffic categorization over Tor.**

***Keywords*—Tor, Malicious Traffic, Intrusion Detection System.**

I. Introduction

Tor is a popular overlay network that provides anonymous communication over the Internet for TCP applications and helps fight against various Internet censorship [1]. Tor has been growing and consists of around 3800 volunteer Tor routers as of July 2013. It serves hundreds of thousands of users and carries terabyte of traffic daily.

Unfortunately, Tor has been abused in various ways. Copyrighted materials are shared through Tor. The black markets (e.g., Silk Road [2], an online market that selling goods such as pornography, narcotics or weapons[1]) can be deployed through Tor *hidden service*. Attackers also run botnet Command and Control servers (C&C) and send spam over Tor. Attackers choose Tor because of its protection of communication privacy, which is achieved in the following way. A user uses source routing, selects a few (3 by default while the hidden service uses a different mechanism [3]) Tor routers, and builds an anonymous route along these Tor routers. Traffic between the user and the destination is relayed along this route. The last hop, called *exit router*, acts as a "proxy" to directly communicate with the destination. Hence, Tor exit routers often become scapegoats and are bombarded with Digital Millennium Copyright Act (DMCA) notices and botnet and spam complaints or even raided by police [4]. These abusing activities prevent potential volunteers from hosting exit routers and hinder the advancement of Tor as a large-scale privacy-enhancing network.

Tor allows manual configuration of IP and port based policies to block potential malicious traffic. However, traffic over Tor has versatile ports such as P2P traffic, making manual configuration a daunting job for common Tor router administrators. Hence, a pressing need is to investigate malicious traffic over Tor. Our research in this paper fills this gap and differs from the existing research efforts, which mainly focus on traffic protocols and applications. For example, McCoy *et al.* [5] reported that web traffic made up the majority of the connections and bandwidth in 2008. Chaaban *et al.* [6] conducted the analysis of the application usage over Tor through deep packet inspection and found that BitTorrent became the first contributor in terms of traffic volume in 2010.

In this paper, we design and implement *TorWard*, which integrates an Intrusion Detection System (IDS) at Tor exit routers for Tor malicious traffic discovery and classification. Our major contributions are summarized as follows.

TorWard can be deployed on *a university campus* while avoiding legal and administrative complaints. It consists of a NAT (Network Address Translation) gateway and a Tor exit router behind the gateway. Tor traffic is routed through the gateway to the exit router so that we can study the outgoing traffic from Tor. The traffic leaving our exit router is redirected into Tor again through the gateway to reliev the university from legal liability. We understand routing exit traffic into Tor incurs a burden on Tor. Nevertheless, this is the only safe way to investigate malicious traffic over Tor in such a sensitive environment. An IDS is installed on the NAT gateway to analyze the exit traffic before it is routed into Tor. We revise the Tor source code and dynamically maintain firewall rules in order not to interfere with non-Tor traffic. Since the use of *TorWard* in early 2012, we have not received any complaints in our experiments, while a lot of administrative complaints were received each day with a bare exit router on campus. We also preform theoretical analysis to demonstrate the effectiveness of *TorWard* and the real-world data matches our theoretical analysis well.

With *TorWard*, we conduct statistical analysis of malicious traffic through Tor. Key observations include: around 10% of Tor traffic triggers the IDS alerts. Alerts are very diverse, raised over both Tor traffic, DoS attack traffic, spam traffic and others. More than 200 malware are discovered from the alerts, including 5 mobile malware, all targeting Android. Although

[1]On Oct. 2 2013, the FBI took down Silk Road.

978-1-4799-3360-0/14/$31.00 ©2014 IEEE

w did not manually filt r out all fals alarms giv n th hug volum of traffic, w giv confirm d xampl s of major thr ats such as botn t traffic and our goal of this pap r is to show th pr ssur of malicious traffic ov r Tor xits and draw a bas lin for futur intrusion d t ction classification and analysis. W also d riv d *traffic protocol and application* statistics, which is larg ly consist nt with th study in [5], [6] whil w now can obs rv traffic from mobil d vic s. To th b st of our knowl dg , this is th first pap r to p rform malicious traffic cat gorization ov r Tor.

Th r st of this pap r is organiz d as follows: W introduc Tor and r vi w r lat d work in S ction II. W pr s nt th system archit ctur for malicious traffic discov ry, syst m s tup, and th or tically analysis to d monstrat th ff ctiv n ss of *TorWard* in S ction III. W conduct a statistical analysis on Tor traffic and inv stigat various al rts and malwar activiti s in S ction IV. W conclud this pap r in S ction V.

II. Background and Related Work

In this s ction, w bri fly introduc Tor and r lat d work.

A. Tor

Figur 1 illustrat s th basic archit ctur of th Tor n twork. It consists of four compon nts: Tor cli nt, onion rout rs, dir ctory s rv rs, and application s rv r. G n rally sp aking, a Tor cli nt installs *onion proxy* (*OP*) that is an int rfac b tw n Tor n twork and cli nts. *Onion routers* (*OR*) form th cor Tor n twork and r lay traffic b tw n Tor cli nt and application s rv r. Th dir ctory s rv rs hold all public onion rout r information. An application s rv r hosts a TCP application s rvic such as a w b. Tor also provid s a *hidden service* to hid th location of s rv rs. *Bridge* is introduc d as hidd n onion rout rs to furth r r sist c nsorship. Without loss of g n rality, w will us Figur 1 as th xampl archit ctur of th Tor n twork in this pap r.

To anonymously communicat with th s rv r ov r Tor, th cli nt downloads onion rout r information from dir ctory s rv rs and choos s a s ri s of onion rout rs to stablish a thr -hop path[2], r f rr d to as *circuit*. Th thr onion rout rs ar known as *entry* (OR1), *middle* (OR2), and *exit* onion rout r (OR3), r sp ctiv ly. Th cli nt can mix multipl TCP conn ctions, r f rr d to as *streams*, ov r a singl Tor circuit.

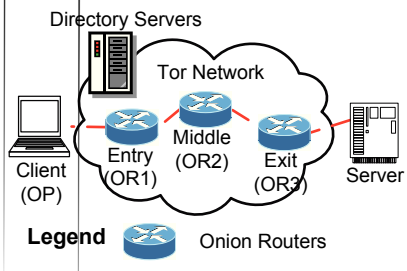


Fig. 1. Tor Network

[2]3 is the default value in Tor.

B. Related Work

Th most r lat d work is [5], [6], which focus s on th n twork protocol analysis to study th *benign* us of Tor. In comparison, our work xplor s malicious traffic ov r Tor.

Malwar authors hav b gan to mak us of Tor to provid anonymous communication b tw n malwar and C&C s rv rs. For xampl , D nnis Brown [7] show d how to configur Z us bot through Tor2W b [8] to conn ct to its C&C s rv r, which is d ploy d as a Tor hidd n s rv r. Malwar Skyn t was discuss d in th w b sit R ddit in 2012 [9]. It d ploys th C&C s rv r as a hidd n s rv r and mb ds a Tor cli nt into th malwar to communicat with th hidd n C&C s rv r. Guarni ri studi d th d tail d f atur s of Skyn t by diss cting malwar sampl s [10], [11]. Two Tor hidd n s rvic bas d malwar s [12], [13] w r r port d in July 2013.

R s arch has b n p rform d to discov r Tor hidd n s rv rs [3], [14]–[17]. For xampl , Øv rli r and Syv rson [14] propos d th pack t counting bas d traffic analysis to id ntify a hidd n s rv r at ntry onion rout rs. Zhang *et al.* [16] l v rag d HTTP f atur s to id ntify a hidd n s rv r at ntry onion rout rs. Murdoch [15] mploy d a clock sk w bas d approach to ch ck wh th r a giv n Tor nod is a hidd n s rv r or not. Ling *et al.* [3] propos d a protocol-l v l bas d hidd n s rv r discov ry approach. Biryukov *et al.* [17] studi d how to d ploy th hidd n s rvic dir ctory to harv st hidd n s rvic information and inv stigat d th pack t counting bas d traffic analysis to locat hidd n s rv rs.

Oth r anonymous communication syst ms w r wid ly abus d as w ll as Tor. For xampl , Tian *et al.* [18] studi d how to trac back th r c iv r who is r tri ving ill gal fil ov r th Fr n t [19].

III. Malicious Traffic Collection

In this s ction, w first pr s nt th archit ctur d sign of *TorWard* to coll ct and analyz malicious traffic in th liv Tor n twork and th n laborat th d tail d syst m s tup. At last, w analyz th ff ctiv n ss of *TorWard*.

A. System Architecture

W cat goriz Tor traffic as inbound and outbound traffic. Inbound Tor traffic is ncrypt d and transmitt d b tw n OR and OR or b tw n OP and OR. Outbound Tor traffic is d crypt d by th Tor xit rout r and forward d to an application s rv r. An xit rout r b hav s as a proxy for a Tor cli nt and communicat s with th application s rv r. Th r for , m dia compani s, ISPs (Int rn t S rvic Provid rs), and campus IT d partm nt may d t ct malicious outbound Tor traffic and dir ct complaints to “off nding” xit rout r a.atnwns I aoTd

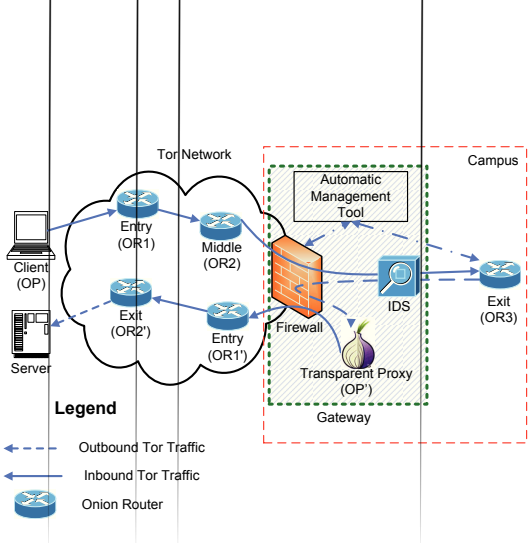


Fig. 2. System Architecture for Malicious Traffic Collection

privat n twork to th campus n twork. Th privat n twork includ s a Tor xit rout r and a Tor cli nt. Port forwarding is nabl d at th fir wall to nabl communication b tw n th xit rout r and middl rout rs in th public n twork. To attract oth r Tor cli nts to s l ct our xit rout r, our xit rout r is s t to acc pt all traffic and has a r lativ ly larg av rag bandwidth and burst bandwidth of $16Mbps$ and $32Mbps$, r sp ctiv ly.

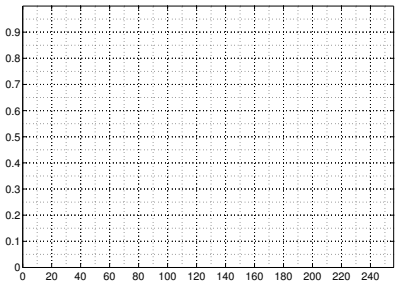
To avoid administrativ and l gal complaints, *TorWard* r di- r cts outbound traffic at our xit rout r into th Tor n twork. W d v lop an automatic manag m nt tool to automatically add and d l t forwarding rul s for th fir wall. W modify th cod of th xit rout r in ord r to s nd th outbound conn ction information (i. ., th d stination IP addr ss and port) to this tool. In particular, b for an xit rout r initiat s an outbound ss

cnformalbo p lo

We can see that $\mathcal{P}_n$ grows significantly as n increases. According to the current Tor router bandwidth real-world data [26], we can calculate the probability $\mathcal{P}_n(b)$ based on the number of circuits n. We set up the bandwidth of our exit router as $16Mb/s$, while the theoretical maximum average bandwidth is $80Mb/s$. Figure 4 illustrates the relationship between $\mathcal{P}_n(b)$ and the number of circuits. It can be observed that the probability $\mathcal{P}_n(16)$ approaches 100% when a malicious Tor client creates around 260 circuits, while the probability $\mathcal{P}_n(80)$ approaches 100% at the Tor exit router with bandwidth $80Mb/s$ after creating around 45 circuits. Consequently, if we have more bandwidth, we can collect malicious traffic more efficiently.

Let $\mathcal{P}_k(b)$ be the probability that a malicious Tor client creates at least one circuit traversing our exit router after establishing k circuits. Assume that $\mathcal{P}_k(b)$ approaches 100%. Let t_i be the average time of creating a new circuit for the i^{th} type of malicious Tor client. We can obtain the average of total time T of retrieving all m malicious traffic by

$$T = max\{t_1 * k, \ldots, t_i * k, \ldots, t_m * k\}. \quad (3)$$



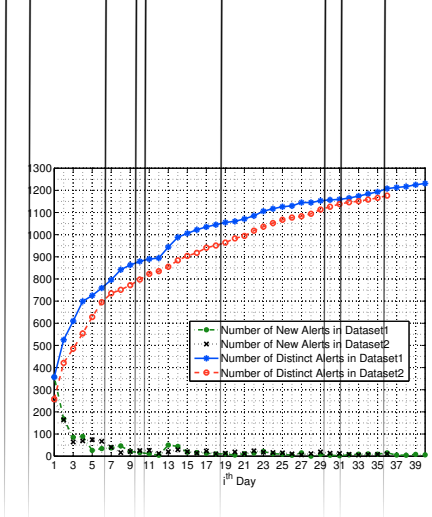


Fig. 5. The relation between number of discovered alerts and ith day

of inbound and outbound Tor traffic. To id ntify inbound Tor traffic, w us TShark's protocol filt r [30] to analyz original traffic and find that around 50% traffic is TLS (Transport Lay r S curity) traffic, which is us d by Tor to ncrypt th inbound Tor traffic. Aft r filt ring th Tor TLS traffic, w mploy a DPI

TABLE V. MALWARE DISCOVERED THROUGH ALERTS

Platform	Classification	Example Malware	Number of Malware
PC	Virus	Win32/Virut.A, Win32/Sality-GR, Win32/Sality.AM, W32/Virut.n.gen, VirTool.Win32/VBInject.gen!DM, Brontok, Luder.B	7
	Worm	Win32.Duptwux/Ganelp, Win32/Fujacks, Win32/Ruskill/Palevo, Win32/Gamarue.F, Win32.AutoTsifiri.n, Win32/Cridex.E, Worm.Win32.Balucaf.A, Koobface Beaconing (getexe), Vobfus/Changeup/Chinky, Win32/Zhelatin, Possible Bobax	11
	Trojan	Win32/Cutwail.BE, Zbot (AS9121), Win32/Tibs, Win32.Farcit.A/Pony, Win32/Sinowal/sinonet/mebroot/Torpig, etc.	88
	Backdoor	Win32/Prosti, Win32/Hupigon.CK, Win32/Bifrose/Cycbot, Win32.Alldibot.A, Win32.Gh0st, Win32/Kbot, etc.	39
	Spyware	Baidu.com Spyware Bar, AskSearch Toolbar Spyware User-Agent, Casalemedia Spyware, Alexa Search Toolbar User-Agent (Alexa Toolbar), ISearchTech.com XXXPorn-Toolbar Activity (MyApp), etc.	45
	Bot	Yoyo-DDoS Bot, JKDDOS DDoS Bot, BlackEnergy DDoS Bot, Illusion Bot, Zeus Bot, P2P Zeus, Darkness DDoS Bot, SpyEye, IMDDOS Botnet User-Agent STORMDDOS, Dropper.Win32.Agent.bpxo, Win32/Dorkbot(NgrBot), Andromeda, Known Skunkx DDOS Bot User-Agent Cyberdog, MRSPUTNIK, ZeroAccess/Sirefef/MAX++/Jorik/Smadow	14
	Adware	W32/OpenCandy, Adware.Gen5, Adware.iBryte.B, Win32.AdWare.iBryte.C, ADWARE/InstallCore.Gen, Win32/InstallMonetizer.AC, Adware.Solimba, AdWare.Win32.Eorezo, BInet Information Install, Adware/Win32.MediaGet User-Agent (mediaget), Common Adware Library ISX User Agent, W32/GameVance Adware	12
Mobile Device	Malware	Android/Qdplugin.A, Android/Adware.AirPush.D, Android.Troj.FakeSms.a, Android/Plankton.P, Android.Plankton/Tonclank	5

Bad-unknown consists of div rs DNS qu ri s and HTTP r qu sts for suspicious domains, such as .co.cc, .tk, .org.pl, .cz.cc, .co.tv, .x .cx, and oth rs. Th s suspicious domains can b us d by C&C s rv rs. W also find al rts form HTTP r dir ction to Sutra TDS (Traffic Dir ction Syst m) that might forc a cli nt to download malwar .

Shellcode-detect al rts indicat that th cont nt of th traffic contains various no op ration (NOOP) strings. Th attack r can s nd long strings of NOOPs to ov rflow th buff r and gain root acc ss to an x86 Linux syst m. W also find h ap spray string r lat d al rts.

Not-suspicious al rts ar for IP addr ss s blacklist d by Abus at.org, Robt x.com and Sorbs.n t for spam mails. B caus Tor xit rout rs may r lay spam mails, th ir IP addr ss s ar also blacklist d and r comm nd d for blocking by thos w bsit s.

Attempted-recon al rts includ th pot ntial SSH port scans. Th al rts sugg st that som Tor cli nts probably att mpt to scan th SSH port. Also, w find th activiti s of r tri ving th xt rnal IP addr ss s of th Tor xit rout r from w b sit s such as showip.n t, myip.dnsomatic.com, cmyip.com, ipchick n.com, whatismyip.com, showmyip.com, and oth rs. Sinc a numb r of malwar s try to g t th xt rnal IP addr ss onc th victim host is inf ct d, th inquiry traffic might b r rout d into th Tor n twork and r lay d by our xit Tor rout r.

Al rts for *attempted-admin* includ thos for a typ of buff r ov rflow vuln rability caus d by a boundary rror in th GIF imag proc ssing of N tscap xt nsion 2. W also discov r d http post r qu sts with n gativ cont nt l ngth that can caus buff r ov rflow at a w b s rv r. Microsoft Dir ctShow AVI fil buff r ov rflow al rts w r found, and this vuln rability allows a r mot attack r to x cut malicious cod at a Tor cli nt.

Web-application-attack al rts ar for two typ s of attacks: attacks from th cli nt sid and attacks from th s rv r sid . W obs rv d that th al rts w r from th cli nt sid , including SQL inj ction attacks by using th Havij SQL inj ction tool. Th al rts from th s rv r sid w r from malwar in th w b pag and cross-sit scripting attacks, which allow th malicious cod to b x cut d by a Tor cli nt brows r.

Attempted-user al rts ar trigg r d by th inbound traffic that att mpts to utiliz various vuln rabiliti s of w b brows rs (.g., Mazilla Fir fox and Microsoft Int rn t Explor r) at th Tor cli nt sid to launch attacks. Th r mot attack r may tak advantag of th s vuln rabiliti s to x cut th malicious cod and control th machin wh r th Tor cli nt is host d. For xampl , w found al rts that r port a r mot s rv r's att mpt to r turn a fil mb dd d with a Class ID (CLSID) to th w b cli nt at th Tor cli nt sid . Th r ar also al rts r lat d to cross-sit scripting (CSS) attacks.

Attempted-dos al rts show that malicious cod is d t ct d in th incoming traffic, xploiting th stack xhaustion vuln rability in th Microsoft Int rn t Explor r Script Engin . If th Tor cli nt us s a vuln rabl v rsion of th w b cli nt to op n th malicious w b pag , th w b cli nt can b t rminat d as a r sult of DoS attacks.

C. Malicious Traffic Statistics

In Figur 5, th two upward curv s show th cumulativ numb r of distinct al rts from datas ts 1 and 2, r sp ctiv ly. Th y incr as v ry slowly aft r s v ral days. Th two downward curv s show th numb r of daily discov r d n w al rts. F w n w al rts ar obs rv d aft r a f w days. Th s r sults match our th or tical analysis in S ction III-C. In a first f w days, w hav captur d most of al rts ov r Tor. Appar ntly, n w malicious traffic hav b n m rging according to Figur 5.

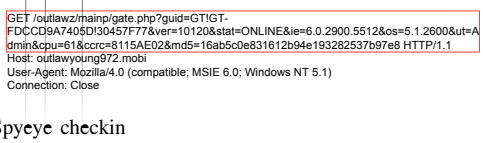


Fig. 6. Spyeye checkin

According to th IDS rul s t classification [40], w cat goriz d th discov r d al rts into s v ral cat gori s. Tabl s III and IV list th numb r of al rts of various groups coll ct d in two datas ts, and d tail d d scription is shown b low. Not that w appli d two distinct IDS rul s ts for th s two datas ts. In Tabl s III and IV, th r ar $8,116,775$ al rts for datas t 1 and $3,624,700$ al rts for datas t 2. Policy-violation al rts hav th larg st p rc ntag , and th y ar incurr d by P2P traffic. Al rts r lat d to malwar includ *unclassified*, *misc-attack*, *trojan-activity*, *not-suspicious*, and *misc-activity*. Th s al rts involv w ll-known or pot ntial IP addr ss s of C&C s rv rs, w ll-known malicious traffic, suspicious DNS qu ry traffic, spam traffic, and suspicious IRC traffic.

To und rstand th traffic volum in diff r nt cat gori s, w calculat th volum of incoming and outgoing traffic from raw data bas d on th al rts. Tabl s VI and VII show th traffic information of datas t 1. Tabl s VIII and IX giv th traffic information of datas t 2. Th r sults ar sort d bas d on th priority of th rul s t [40]. From Tabl s VI and VII, w can obs rv that around $(16GB + 375GB) = 391GB$ out of $3.95TB$ traffic, i. ., around 10% traffic in datas t 1, can trigg r th al rts. Mor ov r, th policy-violation traffic is th most dominant on , which was mainly caus d by P2P traffic. Th n, th s cond dominant traffic is trojan-activity traffic. In addition, th volum of traffic g n rating high priority al rts is much larg r than th volum of oth r traffic.

Bas d on th div rs al rts, w conclud that outbound Tor traffic consists of num rous malicious traffic, which may pot ntially incriminat th party who hosts a Tor xit rout r. In addition, third-party plug-ins of various brows rs us d by som Tor us rs may l ak th ir privat information. In th following subs ction, w furth r xplor th issu s incurr d by malwar activiti s to r v al th impact of th malicious traffic.

TABLE VI. INCOMING TRAFFIC STATISTICS (DATASET 1)

Classification	Size (Bytes)	Percentage	Priority
Shellcode-detect	3,880,691,862	24.36%	High
Policy-violation	350,720,319	2.20%	High
Trojan-activity	147,539,256	0.93%	High
Web-application-attack	18,419,483	0.12%	High
Attempted-user	4,974,891	0.03%	High
Attempted-admin	683,681	0.004%	High
Bad-unknown	155,986,606	0.98%	Medium
Misc-attack	11,186,217	0.07%	Medium
Attempted-recon	174	0.000001%	Medium
Misc-activity	46,947,883	0.29%	Low
Not-suspicious	35,000,208	0.22%	Low
Network-scan	1,018	0.000006%	Low
Unclassified	11,278,737,267	70.80%	Unknown
Total	15,930,888,865		

TABLE VII. OUTGOING TRAFFIC STATISTICS (DATASET 1)

Classification	Size (Bytes)	Percentage	Priority
Policy-violation	370,283,402,491	98.70%	High
Trojan-activity	3,932,683,916	1.05%	High
Attempted-user	142,657	0.00003%	High
Web-application-attack	122,106	0.00003%	High
Bad-unknown	730,210,376	0.19%	Medium
Attempted-recon	72,195,354	0.02%	Medium
Not-suspicious	80,125,728	0.02%	Low
Misc-activity	50,643,140	0.01%	Low
Total	375,149,525,768		

TABLE VIII. INCOMING TRAFFIC STATISTICS (DATASET 2)

Classification	Size (Bytes)	Percentage	Priority
Shellcode-detect	238,734,330	9.55%	High
Trojan-activity	220,219,632	8.81%	High
Policy-violation	66,546,599	2.66%	High
Web-application-attack	16,888,851	0.68%	High
Attempted-user	70,334,933	2.81%	High
Attempted-admin	48,477,268	1.94%	High
Misc-attack	552,748,810	21.40%	Medium
Bad-unknown	200,336,881	8.02%	Medium
Web-application-activity	595	0.00002%	Medium
Attempted-recon	178	0.000007%	Medium
Misc-activity	534,901,914	22.12%	Low
Not-suspicious	30,105,913	1.20%	Low
Protocol-command-decode	8,124,809	0.33%	Low
Network-scan	1,253	0.00005%	Low
Unclassified	511,917,854	20.48%	Unknown
Total	2,499,339,820		

D. Malware Activities

As shown in Tabl V, w discov r d various activiti s associat d with malwar s from th r port d al rts, including th communication b tw n malwar and C&C s rv r, DoS attacks, Spams, and oth rs.

Communication between Malware and Command and Control Server: Som malwar s ar d sign d to conn ct to a C&C s rv r in ord r to r port th information r tri v d from th victim machin , updat th malwar , download th configuration fil , and p rform oth r op rations. To hid th communication b tw n malwar and th C&C s rv r, malwar authors may adopt Tor to hid malicious traffic and prot ct th r al location of th C&C s rv r from b ing discov r d. If th malwar choos s our Tor xit rout r, th malicious traffic will trav rs th Tor circuit and stablish th conn ction to th C&C s rv r through our xit Tor rout r. Th r for , our xit Tor rout r can d t ct such malicious traffic. In datas t 2, w discov r d 622 C&C s rv r IP addr ss s bas d on ch ck-in m ssag s from mor than 70 diff r nt known malwar , 59 diff r nt IP addr ss s of known compromis d or hostil hosts that might b d ploy d as a C&C s rv r, 71 C&C S rv r IP addr ss s r port d by Shadows rv r [41], and 93 IP addr ss s obtain d from various w ll-known track rs (.g., Z us, Spy y , and Pal vo track rs) that r port C&C s rv rs' IP addr ss s.

W now show a f w xampl s found in our datas ts on how malwar s communicat with th ir C&C s rv rs. In Figur 6, a Spy y bot is conn cting to its C&C s rv r to r port th information of th victim machin . According to th

TABLE IX. OUTGOING TRAFFIC STATISTICS (DATASET 2)

Classification	Size (Bytes)	Percentage	Priority
Policy-violation	159,692,890,115	98.52%	High
Trojan-activity	2,004,499,807	1.24%	High
Attempted-user	39,242	0.00002%	High
Web-application-attack	7,631	0.000005%	High
Bad-unknown	174,978,188	0.11%	Medium
Attempted-recon	35,503,428	0.02%	Medium
Attempted-dos	5,373	0.000003%	Medium
Not-suspicious	166,439,974	0.10%	Low
Misc-activity	10,796,084	0.007%	Low
Protocol-command-decode	4,586,076	0.003%	Low
Total	162,089,745,918		

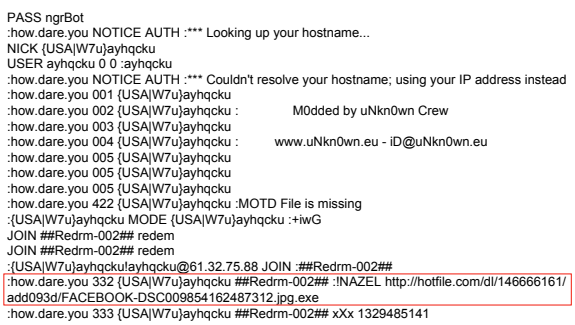


Fig. 7. Ngrbot

format of SpyEy C&C M ssag [42], w can pars th information that includ s a uniqu id ntifi r (guid=GT!GT-FDCCD9A7405D!30457F77), th v rsion of th bot inf ctor (v r=10120), th status of th bot (stat=ONLINE), th v rsion of Int rn t Explor r (i =6.0.2900.5512), th v rsion of Microsoft Windows op rating syst m (os=5.1.2600), th typ of th curr nt us r on th victim machin (ut=Admin), th CPU load (cpu=61), th CRC32 tak n from th last four byt s of th bot configuration fil (ccrc=8115AE02), and th md5 of th bot inf ctor (md5=16ab5c0 831612b94 193282537b97 8). Figur 7 shows that a Ngrbot logs into a IRC s rv r, joins a chat room and th n r c iv s a command to download anoth r malwar . W found malicious traffic from mobil d vic s as w ll. As an xampl , Figur 8 illustrat s th malwar communicating with th r mot s rv r by using HTTP protocol.

DoS Attacks: A bot mast r can control a larg numb r of bots and malwar to p rform a DoS attack through Tor. For xampl , in our m asur m nts, w discov r d 72, 894 DoS attack al rts of Yoyo-DDoS bot wh r 457 distinct d stinations ar found. Yoyo-DDos bots can r c iv th command of attacking a targ t s rv r from th bot mast r and th n continuously s nd HTTP r qu sts to th targ t s rv r so as to launch HTTP flood attacks. Th targ t s rv rs of 96% DDoS attacks that w found ar locat d in two countri s, th Unit Stat s and China.

Spam Traffic: W found 40, 834 r lat d spam al rts and 8, 186 distinct mail s rv r IP addr ss s from 115 diff r nt countri s in datas t 2. As w can s from Tabl X, 89.02% al rts originat from only 10 countri s, whil around 50% mail s rv rs ar from only thr countri s. Du to th larg numb r of spams from Tor n twork, many mail s rv rs d ny th mail r lay d from th Tor n twork. This hurts b nign Tor

into Tor. We analyze the data collected over a long period and discover that Tor carries a large amount of malicious traffic, including various P2P, botnet, spam, and other malware traffic. Among the $3,624,700$ alerts raised in one of our datasets, 78.03% of them are caused by P2P traffic, while 8.99% are related to malwares. As an ongoing work, we are conducting research on blocking and sanitizing malicious traffic at Tor exit routers and contributing to the healthy development of Tor.

ACKNOWLEDGMENTS

This work was supported in part by National Key Basic Research program of China under grants 2010CB328104, China National High Technology Research and Development Program under Grants No. 2013AA013503, National Natural Science Foundation of China under grants 61272054, 61202449 and 61320106007, Natural Sciences and Engineering Research Council of Canada (NSERC), by US NSF grants 1116644, 0942113, 0958477, 0943479, and 1117175, China National Key Technology R&D Program under Grants No. 2010BAI88B03 and 2011BAK21B02, by China Specialized Research Fund for the Doctoral Program of Higher Education under grants 20110092130002, Science Research Foundation of Graduate School of Southeast University, Jiangsu Provincial Key Laboratory of Network and Information Security under grants BM2003201, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under grants 93K-9. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do