SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2012; **5**:572–582
Published online 20 July 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.352

RESEARCH ARTICLE

Anti-virus in-the-cloud service: are we ready for the security evolution?

Wei Yan[1] and Nirwan Ansari[2]*

[1] HS USA, Cupertino, CA 95014, U.S.A.
[2] New Jersey Institute of Technology, Newark, NJ 07102, U.S.A.

ABSTRACT

The ever-increasing malware variants pose serious challenges for traditional signature-based anti-virus (AV) scan engines. To effectively handle the scale and magnitude of new malware variants, AV functionality is being moved from the user desktop into the cloud. AV in-the-cloud service is becoming the next-generation security infrastructure designed to defend against virus threats. It provides reliable protection service delivered through data centers worldwide, which are built on virtualization technologies. Nowadays, cloud-based security services are gaining bullish projections in both consumer and enterprise markets. However, are we getting ready for the cloud evolution? Security vendors are facing various challenges regarding the architectural design, implementation, and validation. Owing to the lack of operation standards among vendors and very few research works conducted up to this point, researchers have no references of AV cloud testing to rely on. In this paper, the architecture of AV in-the-cloud service is described. The challenges and solutions are discussed and illustrated by examples taken from our cutting-edge research on practical applications. Copyright © 2011 John Wiley & Sons, Ltd.

KEYWORDS
anti-virus; malware pattern; in-the-cloud; network security

***Correspondence**
Nirwan Ansari, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, U.S.A.
E-mail: nirwan.ansari@njit.edu

1. INTRODUCTION

With the popularity and variety of zero-day malware over the Internet, generating their signatures to detect them via anti-virus (AV) scan engines becomes an important reactive security function [1]. Security engineers are facing a serious problem of defeating the complexity and quantity of malware. For example, they have to keep on inserting new virus signatures into the database. This trend of increasing the size of the signature database required to execute security applications consume much of the PC memories and resources. As a consequence, customers always complain that security software bog down their computers.

Anti-virus in-the-cloud service has been advocated as the next-generation model for virus detection by Trend Micro (www.trendmicro.com) since June 2008. It is a software distribution model in which security services are hosted by vendors and made available to customers over the Internet. This approach employs a cloud server pool that analyzes and correlates new attacks and generates vaccinations online. The cloud infrastructure will sharply reduce computation burdens on the clients and enhance security products in mitigating new malware. Furthermore, customers only need to maintain a small and lightweight version of a virus signature file instead of the full copy. Benefits include easy deployment, low costs of operation, and fast signature updating. Currently, major security vendors, such as Trend Micro, Symantec (www.symantec.com), F-secure (www.fsecure.com), and McAfee (www.mcafee.com), are all developing cloud security products to handle the exponentially growing volume of malware.

For a suspicious file, the AV desktop agent fetches the fingerprint or calculates the hash value of the file and sends it to the remote cloud server, which will compare that fingerprint or value with the continuously updated signature database in the Internet. If the value exists in the database, the client will be asked on which specific action the user wants the desktop agent to take on the infected file. For example, a user can quarantine, block, or clean the detected malicious file.

1.1. Threat response system

A significant increase in the spread of viruses, worms, and Trojans over the Internet has been observed during the past few years. Traditional threat response systems involve malware collection, signature generation, and then signature database updating. However, owing to the flood of malware, security companies usually receive thousands of suspicious samples daily from honeypots and customers' submissions. It is very time consuming and resource intensive for them to analyze these samples manually and generate the signatures.

In AV in-the-cloud, users have the option to send new suspicious files found by desktop agents to the threat response system for analysis. If necessary, a new signature will be created to detect that file or the malware family to which the file belongs. Some traditional commercial security products, for example, Panda's TruPrevent (www.pandasecurity.com) and McAfee's Artemis (www.mcafee.com/artemis), have included this functionality; there are still many "spikes" of high virus-scanning latency that cannot be ignored. Last but not least, temporal changes in file size, file type, and storage capacity in modern operation systems are slowing down virus scan. However, with the large-scale deployment of in-the-cloud desktop agents, it is doubtless to say that threat response systems will receive many more malware samples every day. Has the AV industry developed the mature techniques to cope with the large data processing required by threat response systems?

To handle the large quantity of new unknown samples, it is important to develop intelligent threat response systems, which support automatic and generic signature generation for both static and heuristic detection. In this paper, we will describe a new malware signature generation scheme, signature-in-cloud (SiC), which has been implemented and tested. This hybrid system combines advantages of prior knowledge of known viruses in traditional AV signature databases and the ability of computational intelligence to detect new unknown malware variants. Our experiments on millions-scale samples show that SiC keeps a good workload balance between the desktop and the cloud server, and its signature generation overhead is much less than that of traditional solutions. Furthermore, SiC achieves superior performances in terms of detection rates and false positives.

1.2. Threat model

Anti-virus cloud services are becoming attractive attack targets because shutting down a cloud server cluster is more ominous than compromising a single machine. Therefore, preventing cloud servers from being compromised has become a critical issue. The attacks can be brute force as well as technically sophisticated. The communication link between a desktop and a cloud server is over the Internet. What will happen if the physical link is severed? On the other hand, to defend against network attacks, cryptography can be used to scramble packet contents. However, by using statistical analysis, an attacker may determine the next hop to which packets may traverse. With this information, the attacker can launch distributed denial of service (DDoS) against the cloud servers to significantly deteriorate the quality of service (QoS).

During the past years, anonymous networks [2] have been used to provide private and secure communications for a variety of applications. One important feature of anonymous networks to fortify AV cloud is the location-hidden service; that is, a server can communicate with a user without revealing its real identity. To build a communication link in an anonymous network, the desktop agent will choose a set of authorized anonymous nodes and incrementally create an encrypted circuit to the cloud server. Because each anonymous circuit is extended one node at a time, a node in the link only knows its immediately previous and following nodes [3,4].

In general, anonymous networks fall into two categories: high-latency and low-latency networks. The high-latency networks like Mixminion (www.mixminion.net) work as a store-and-forward relay mix to mitigate global adversaries and strong traffic analysis attacks. However, a big drawback of the high-latency system, as its name suggests, is that it will introduce long delivery delays. As a result, high-latency anonymous networks are generally used in the high-latency communication systems like anonymous emails. On the contrary, low-latency anonymous networks, such as Tor [2], are suitable for interactive applications such as web browsing and online chatting. In AV in-the-cloud service, the communication between a desktop and a server over the Internet, including transmitting a virus-scan request and returning the requested result, usually requires as little as hundreds of milliseconds, that is, especially low latency. Therefore, in this paper, we investigate the traffic analysis attacks and their threats against the quality of anonymity (QoA) in low-latency anonymous communication networks.

Murdoch and Danezis [3] and Bauer *et al*. [4] showed that low-cost attacks were highly effective at compromising the end-to-end anonymity of Tor. In a previous work, Borosiv *et al*. [5] demonstrated that the selective denial of service can also reduce the anonymity considerably. In this study, we consider the low-cost attack model in which an adversary can only observe part of the AV in-the-cloud network. By compromising the user's desktop and a few selectively anonymous nodes, a non-global adversary can apply DDoS to significantly delay packets traversing the anonymous network and to deteriorate QoA, thus exposing the systems to traditional network attacks.

At the time of this writing, few, if any, research findings have been reported on mitigating attacks on the AV in-the-cloud infrastructure, and very few open discussions have been discoursed in the community. Although some results on applying traffic analysis attacks on Tor [3,5] have been reported, this paper, to our best knowledge, is the first to emphasize on protecting the communication links between the agents and the cloud servers. Furthermore, about 20 million samples, the largest dataset among related research works, have been used to evaluate our intelligent threat response system. The rest of the paper is organized as follows. We begin in Section 2 by describing the infrastructure of AV in-the-cloud service, followed by the description of the novel desktop agent solution. Section 3 describes SiC, an automatic and generic virus signature generation scheme for modern threat response systems. Traffic analysis attacks and the corresponding defense solutions are discussed in Section 4. Section 5 presents the conclusion.

2. ANTI-VIRUS IN-THE-CLOUD SERVICE

The new malware variants challenge the traditional AV protection model, which demands frequent signature updates, large signature databases, and resource-guzzler style security products. As the next-generation security infrastructure, AV in-the-cloud service is moving the virus-scanning functionality from the desktop to the Internet.

2.1. Traditional anti-virus solutions

Computer viruses a.k.a. malware are malicious programs used to compromise computers and steal confidential data. Most malware are executable files that can be understood and executed by operating systems. For example, portable executable (PE) format [6] is the most common executable format for Windows. A PE file comprises various sections and headers that describe the section data, import table, export table, resources, and so forth. To search a PE file for malware, an AV scanner typically scans the original entry point, the execution entry point of the PE file, for known signatures. PE tools, such as IDA Pro (Hex-Rays, Boulevard de la Sauvenière, 30 4000 Liège, Belgium) (www.datarescue.com) and Ollydbg (www.ollydbg.de), facilitate the ease to view and analyze WIN32 PE files.

A traditional AV scanner is deployed at the desktop. For any suspicious file, the scanner searches for the file's signature or hash value in the signature database. Traditional signature database usually employs prior knowledge of malware signatures, which are generated by security engineers using reverse-engineering techniques [1]. The signature database is efficient to detect known malware with low false positive rates. However, it cannot often detect unknown viruses and polymorphic variants. Polymorphic malware can mutate their signatures, via unpredictable compression or encryption transformations, and easily bypass AV scanners. In order to detect these polymorphic malicious programs, AV scanners have to estimate all possible forms that these programs may mutate. As a result, security engineers have to keep on inserting new virus signatures into the database whenever a new threat occurs. This trend for ever larger signature files required by security products consume much of the PC memory and resources, thus slowing down computers significantly. Furthermore, some customers are agitated when the automatic signature updating happens several times a day.

On the other hand, conventional signature generation solutions always require heavy manual involvement by studying the emulation traces with hours or even days of delay. Generating signatures for zero-day threats becomes a tedious reactive security function. Security vendors are facing great challenges in overcoming the complexity of malware, and fighting against the malware backlog is nothing new.

2.2. Anti-virus cloud infrastructure

Figure 1 shows the architecture of AV in-the-cloud service. The agent is an on-access scanner deployed at the desktop. It places itself between the applications and the operating system. The agent automatically examines the local machine's memory and file system whenever these

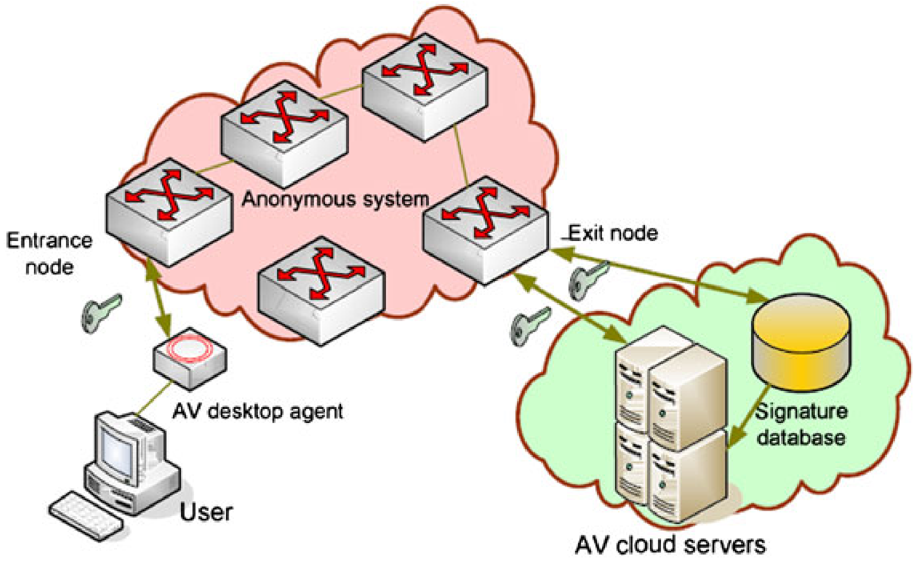


Figure 1. Anti-virus (AV) in-the-cloud infrastructure.

DOI: 10.1002/sec

resources are accessed by an application. For any suspicious file, the agent generates the hash value or a specific signature of the file and sends it to the remote cloud server for security verification. The low-latency anonymous communication network is used to forward these requests from the desktop to the remote cloud. By distributing a set of trusted anonymous hops, it offers the location-hidden service without revealing the cloud server's networking identity.

In the anonymous system, the virus-scan-request packets are routed through anonymous nodes on their circuits towards the cloud servers. A circuit is built from the agent one step at a time. After the whole circuit is set up, the first node in the path is called the entrance node, and last node is the exit node. Data in the anonymous network are encrypted with a layered encryption scheme. When a packet reaches the entrance node, the node decrypts the data and the routing information of the next hop [2]. This process is repeated until the packet reaches the cloud server via the exit node. Afterwards, the server retrieves the decrypted hash value or fingerprint in the signature database and sends the detection result back to the agent.

Figure 2 shows the architecture of the AV cloud agent. The cloud agent is a lightweight hybrid desktop solution to resolve the AV resource-intensive problem. It acts like a file filter, inspecting suspicious file loading and storing activities. The agent collects hash values or fingerprints of suspicious files from users. These users can be either individually distributed or locally networked. If the hash values or fingerprints are already stored in the cache, the agent just returns the cached results to inform the users whether the requested files are malicious or not. Otherwise, it will search in the local lightweight signature database or directly send the values or fingerprints into the cloud.

Figure 3 shows two kinds of clouds for a cloud-based datacenter: public cloud and private cloud. The public clouds are run by service providers, such as Amazon, Google, and Microsoft. Their services are accessible via internet and suitable for applications, which require non-critical service level agreements (SLAs). On the other hand, the private cloud is suitable for security applications utilized in a secure QoS environment. For example, AV scanning in-the-cloud service and web reputation systems are not suitable for the public cloud because the critical SLAs cannot be met by the public cloud. Other issues include how to efficiently manage AV and web reputation services in the private cloud, how to reduce costs through services provided by economic public cloud providers, and how to guarantee service performance and scalability through SLAs [7].

In order to keep a good workload balance between the desktop and the cloud server, the agent requires a lightweight signature database with size much smaller than that of the traditional one. When a suspicious file cannot be verified by the desktop signatures, the agent will send a virus-scan-request to a cloud server, thus saving the bandwidth without sending too many packets in the cloud.

Nowadays, to evade malicious content detection, virus hackers use binary tools to instigate code obfuscation, which has become the most common method to bypass the security products. Therefore, it is vital for AV products to deploy the emulator to inspect hidden payloads. An emulator includes programs to execute or emulate suspicious encrypted executables until they are fully decrypted in memory. There are two ways to deploy the emulation functionality: an emulator can be embedded inside the desktop agent or deployed in the cloud. An agent without the emulator can relieve users from the resource constraints of desktop virus scanning. However, the agent has to send the full obfuscated samples through anonymous nodes towards the cloud servers. This transmission scenario will consume tremendous amount of bandwidth and is not suitable for customers who have the bandwidth limitation. On the other hand, embedding the emulator into the desktop allows the agent to inspect the hidden payloads of the obfuscated programs. Bandwidth will be saved because hash value of the dumped data rather than the file itself is sent to the cloud.

3. SIGNATURE-IN-CLOUD THREAT RESPONSE SYSTEM

Traditional signature databases are usually generated by tedious reverse-engineering techniques. They are efficient

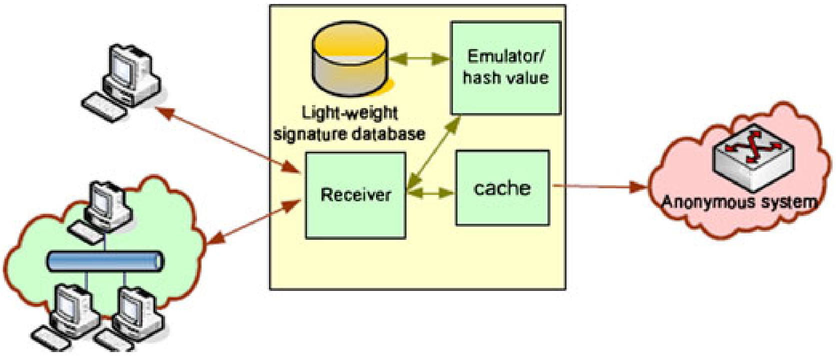


Figure 2. Anti-virus cloud agent architecture.

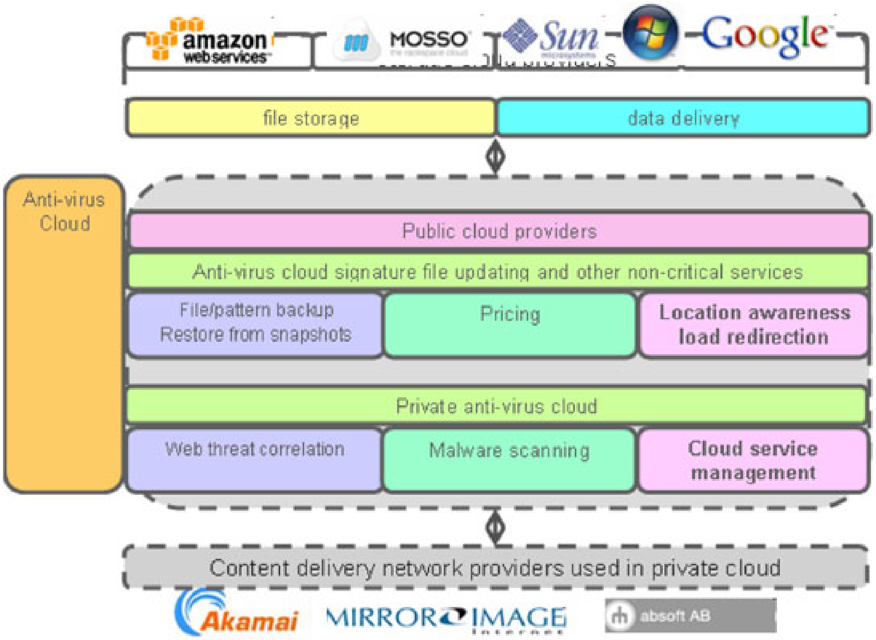


Figure 3. Cloud datacenter infrastructure.

to detect known malware but cannot often catch unknown virus variants. Another drawback is that each virus signature in the database is generally not generic and can only detect one individual malicious program. In order to handle the overwhelming virus-scan requests from desktop agents in time and keep the signature database "in shape" at the same time, SiC has been developed to work with existing signature databases as a core module of the modern threat response system. Within a few minutes, SiC can generate dozens of intelligent and generic signatures for each whole malware family. These signatures including both static and dynamic fingerprints are efficient in mitigating code obfuscation and detecting new malware variants.

Our assumption is based on the fact that samples of the same malware family must have some invariant raw binary sequences. SiC signatures are generated from those sequences. Our utility tool can parse a PE file and list its internal structures, such as PE header, optional header, section table, import table, export table, and the resources.

To build static signatures, the intelligent parser in SiC firstly extracts PE semantic information of samples and creates binary streams. Afterwards, computational intelligence techniques are applied to find feature sequences. Based on these sequences, static signatures are generated for online matching. SiC can also automatically generate behavior signatures for heuristic detection. Nowadays, the emulator or sandbox is used to capture behavior traces. Figure 4 shows the procedure of generating behavior signatures. A malware emulator and its tailored malicious behavior ontology are used to collect dynamic execution token traces. Afterwards, the behavior signatures are generated from these token streams. SiC signatures are compatible with existing virus signature formats and easily integrated into commercial scan engines. Our test results show that SiC can shrink a traditional signature database by many folds. Also, it can improve detection and false positive rates for modern malwares. Figure 5 shows an example of a SiC signature. Concatenating the second column of the figure will give rise to a SiC signature.

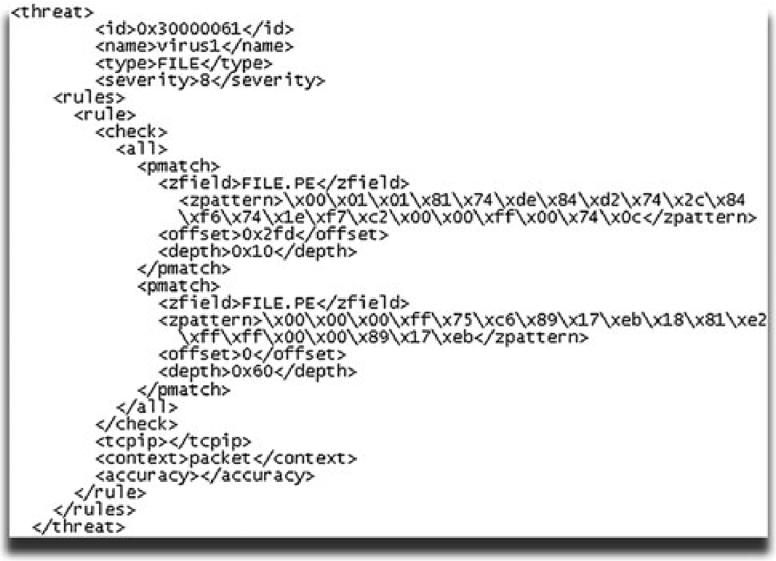
Normally, security researchers take a few hours to manually generate a malicious signature for a new piece of malware. This time interval is no longer acceptable nor scalable because of the exponential increase in malware. The SiC extraction feature helps shorten the response time by automatically extracting signatures of unknown viruses without the need for human intervention. Table I shows the simulation results of processing time to extract signatures from malicious samples.

In our test, a total of 256 smart patterns are selected (22 malware families and 6 PE packer families). An example of three SiC signatures is shown as follows:

(1) 1750238f5b8000000001bc05f83d8ff5ec385c074c48b168b0f38ca75e74874.
(2) 83c40c89450ceb3383fbff74168bc38bcbc1f80583e11f8b0485.
(3) d04c88b0b89088a4d00884804474583c3043bfe7cba33dba1.

Based on the SiC signature, the detection rule of malware will be generated as follows:

Malware families:
onlineGame
Troj.5
HUI.PIGEON2 DIAL.DIALMIN



```
<threat>
        <id>0x30000061</id>
        <name>virus1</name>
        <type>FILE</type>
        <severity>8</severity>
    <rules>
      <rule>
        <check>
          <all>
            <pmatch>
              <zfield>FILE.PE</zfield>
                <zpattern>\x00\x01\x01\x81\x74\xde\x84\xd2\x74\x2c\x84
                \xf6\x74\x1e\xf7\xc2\x00\x00\xff\x00\x74\x0c</zpattern>
              <offset>0x2fd</offset>
              <depth>0x10</depth>
            </pmatch>
            <pmatch>
              <zfield>FILE.PE</zfield>
              <zpattern>\x00\x00\x00\xff\x75\xc6\x89\x17\xeb\x18\x81\xe2
                \xff\xff\x00\x00\x89\x17\xeb</zpattern>
              <offset>0</offset>
              <depth>0x60</depth>
            </pmatch>
          </all>
        </check>
        <tcpip></tcpip>
        <context>packet</context>
        <accuracy></accuracy>
      </rule>
    </rules>
  </threat>
```

SAPIR TROJ.SAMLL BANCOS
SPYBOT
SPYBOT.GEN
Bifrose
NUWAR PornDialer
Trojan.Win32.Dialer Troj.Bank Backdoor.Win32.visel
Adware.Ejik Win32.Nethief Adware.Vapsup
Adware.NaviPromo Adware.Admoke Adware.One-Step
Adware.Boran

PE packer families:
Armadillo
UPack
Themidal v1.7.3
Nspack
VMprotector
ASPack

The following features of signature extraction were defined to describe the performance:

- Minimum signature length.
- Number of sub-signatures.
- Signature offset range.
- Number of signatures.
- Hash value (optional).

Performance results of our threat response system are illustrated below. Our datasets include more than 17 million benign samples and 28 000 samples from 30 malware families. By using the intelligent technique to scan floating relative addresses instead of raw addresses, SiC produces high detection rate; by using disjoint feature segments, SiC achieves low false positive rate; by using back up signatures, SiC can defeat code obfuscation efficiently. The false positive rate of the whole 17 million benign samples is only 0.0006% (1 false positive out of every 150 000 benign samples). With respect to the real-time scanning speed, SiC is comparable with all commercial security products. For example, SiC takes 46 s to scan Windows folder, which includes 10 400 files.

We also measured the detection rate for 30 sample families, which include 10 PE packer and 20 malware families. The training set was chosen from 30 to 50 samples for each family, and the generated SiC signatures were used to detect the rest of the samples for each family. The average malware detection rate is around 80%, and

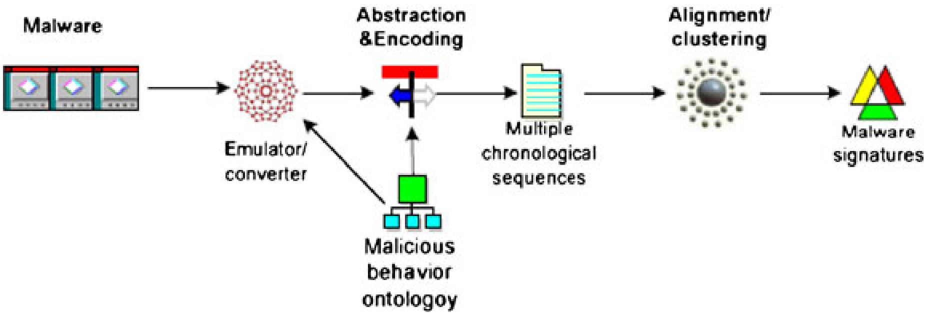
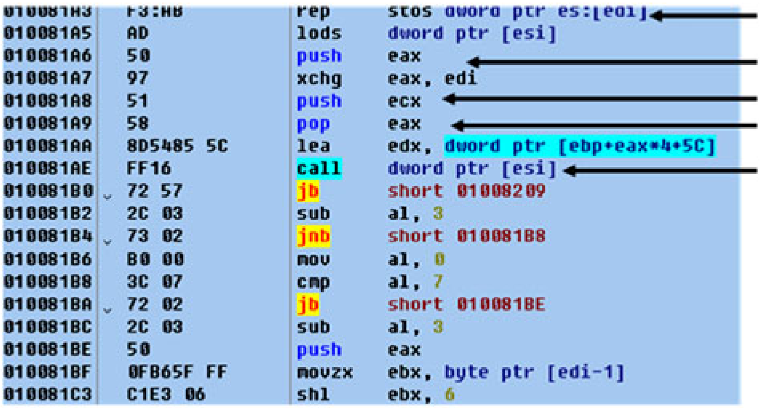


Figure 4. Signature-in-cloud heuristic signature discovery.



```
010081A3   F3:AB       rep    stos dword ptr es:[edi]
010081A5   AD          lods   dword ptr [esi]
010081A6   50          push   eax
010081A7   97          xchg   eax, edi
010081A8   51          push   ecx
010081A9   58          pop    eax
010081AA   8D5485 5C   lea    edx, dword ptr [ebp+eax*4+5C]
010081AE   FF16        call   dword ptr [esi]
010081B0   72 57       jb     short 01008209
010081B2   2C 03       sub    al, 3
010081B4   73 02       jnb    short 010081B8
010081B6   B0 00       mov    al, 0
010081B8   3C 07       cmp    al, 7
010081BA   72 02       jb     short 010081BE
010081BC   2C 03       sub    al, 3
010081BE   50          push   eax
010081BF   0FB65F FF   movzx  ebx, byte ptr [edi-1]
010081C3   C1E3 06     shl    ebx, 6
```

- f3abad509751588d54855cff1672572c03

Figure 5. A signature-in-cloud signature.

Table I. Signature-in-cloud signature extraction time.

Malware content size (mb)	Segment size	Processing time (s)
53.8>>>	0 × 200	10 452
55.0	0 × 100	5439
52.3	0 × 90	3983
45.0	0 × 40	4805
41.0	0 × 60	4108

most of the remaining 20% samples can be detected by traditional signatures (for a larger training set, the detection rate will be higher).

Figure 6 presents the detection rates and the number of intelligent signatures for each family. An interesting result was observed: the number of signatures for each packer family is much less than that of the malware family. One reason can be attributed to the fact that fewer signatures are required to capture the unpacking semantics of packer families. Normally, a packer's unpacking process involves four consecutive steps: decompression or decryption, anti-debugging checks, import table rebuilding, and jumping to Original Entry Point (OEP) [6]. For each packer family, each of the aforementioned steps always involves a similar work flow. Even if it is well known in the security industry that packers are more tricky and complicated than common malwares, our SiC threat response system is very efficient in detecting the packed samples.

Figure 7 presents the traffic simulation performance of file payload scanning. A total of 300 signatures were used, and 1279 HTTP connections were attempted. The scanning speed is 802.651 Mbps, which is above the average industry performance.

As shown in Figures 8 and 9, data centers deploy virus signature updates on different storage cloud providers and use generic cloud interface for dynamical resource allocation. The functionalities, such as budget constraints, network latency, and load direction, will choose the best storage cloud candidate to match the user's SLA targets, such as throughput, response time, and lowest service costs. On the other hand, data centers maintain back-end correlation system by using Content Data Center's (CDC) rapid and reliable services for malware protection and signature generation. For example, AV vendors generate new virus signatures and store them at different storage providers across the world, such as Amazon and Google. If a user wants to download a new signature update, it firstly sends a request to Trend's Smart Protection Network (SPN) validation module, which will check the reputation of the user's ID or IP address. Afterwards, the local proxy agent will forward the request to the data center. Based on

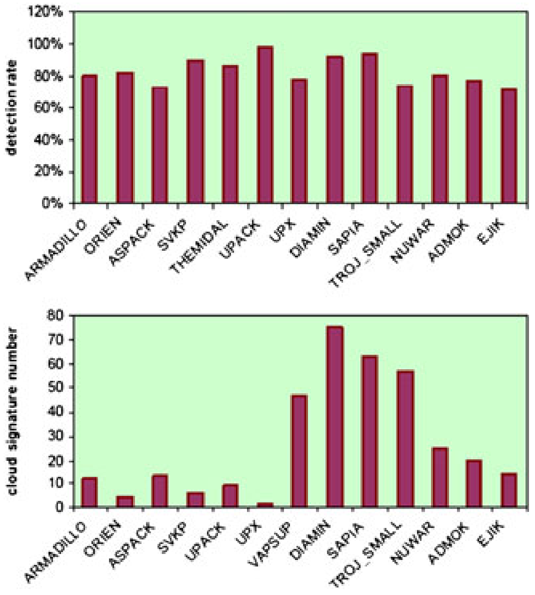
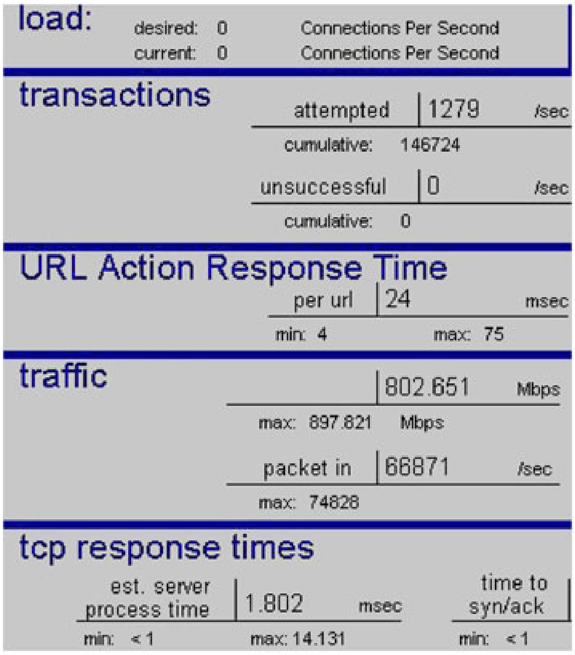


Figure 6. Detection rate and the number of signatures for each malware family.



load: desired: 0 Connections Per Second
current: 0 Connections Per Second

transactions
attempted | 1279 /sec
cumulative: 146724
unsuccessful | 0 /sec
cumulative: 0

URL Action Response Time
per url | 24 msec
min: 4 max: 75

traffic
| 802.651 Mbps
max: 897.821 Mbps
packet in | 66871 /sec
max: 74828

tcp response times
est. server process time | 1.802 msec
min: < 1 max: 14.131
time to syn/ack |
min: < 1

Figure 7. Traffic simulation performance. TCP, transmission control protocol; URL, uniform resource locator.

location awareness, the provider selector evaluates transmission delay, cost, and throughput parameters, and informs the user the best provider candidate to achieve the best SLA.

4. ATTACKS ON ANTI-VIRUS IN-THE-CLOUD

The dramatic expansion of networking applications imposes network security a pressing issue. As increasingly more desktop agents are connected to the cloud, their vulnerabilities easily facilitate an adversary to initiate attacks.

4.1. Global or non-global attack

In cloud networking, all the traffic is encrypted, and an eavesdropper cannot easily access the packet contents. However, the communication links are still vulnerable to traffic analysis attacks. Traffic analysis is a means to extract and infer information, such as packet timing and lengths, without the knowledge of the content payloads. Two kinds of attack scenarios are considered: global attack

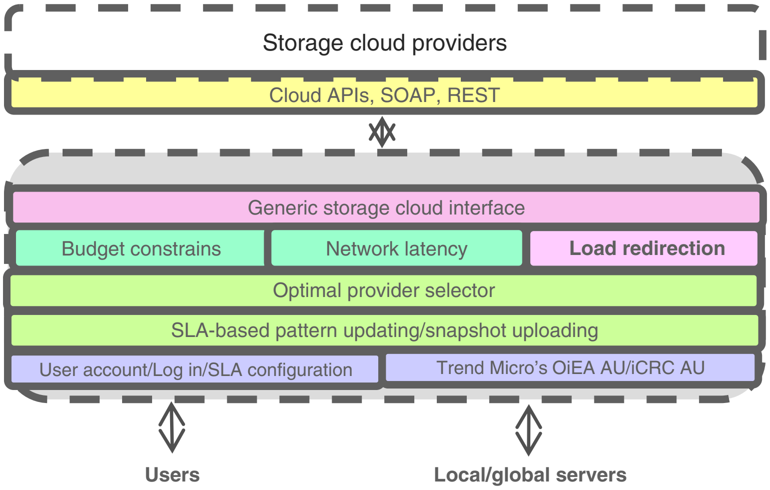

Storage cloud providers
Cloud APIs, SOAP, REST
Generic storage cloud interface
Budget constrains
Network latency
Load redirection
Optimal provider selector
SLA-based pattern updating/snapshot uploading
User account/Log in/SLA configuration
Trend Micro's OiEA AU/iCRC AU
Users
Local/global servers

Figure 8. Signature file update on public storage clouds. APIs, Application Programming Interface; SLA, service level agreement; SOAP, Simple Object Access Protocol; REST, Representational State Transfer.

DOI: 10.1002/sec

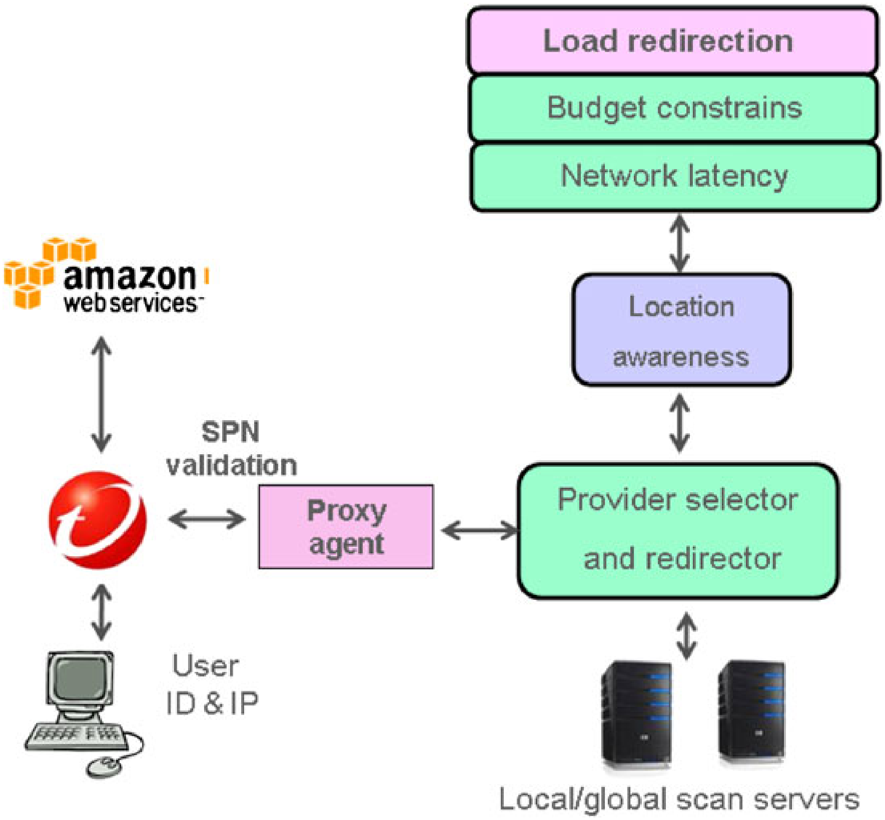


Figure 9. Signature update process. SPN, service provider name.

and non-global attack. In the global attack model, an adversary is assumed to have the capability to observe the whole network. The attacker can use the statistical disclosure attack [8] to trace the origin and the destination of a communication channel. Nowadays, security vendors usually deploy cloud servers and virus signature databases located at data centers worldwide. Adversaries only have the capabilities to observe a fraction of the network but not the totality. Therefore, we only consider how to protect AV cloud service from non-global attackers.

A recent work in [3] described a low-cost attack on Tor by a non-global adversary. By sending the probing traffic through a compromised node and monitoring the traffic latency, the attacker can infer which of the nodes are being used to relay packets. In AV cloud service, virus scanning requires millisecond-scale communication between an agent and the server. Therefore, we only consider the low-latency anonymous network. However, because a low-latency network system cannot significantly distort the traffic timing, it is still vulnerable to the traffic analysis attack.

4.2. Weak-entrance-node problem

In order to attack the cloud service, an adversary requires at least two essential conditions: accessing anonymous nodes and inserting the probing traffic into the communication links. In this section, a vulnerability of the AV in-the-cloud architecture, referred to as the "weak-entrance-node" problem, is discussed. By taking advantage of this vulnerability, the low-cost traffic analysis attack can be carried out by a non-global adversary who merely controls a desktop agent.

A non-global attacking strategy depends on which part of the whole network can be observed or controlled. If an eavesdropper can monitor the virus-scan packets from the desktop agent to the anonymous network edge, it can easily locate the entrance anonymous node. This can be achieved by firstly compromising the desktop or just purchasing the commercial software of the desktop agent and installing it in the local machine. By setting up a sniffer on the desktop and analyzing the network traffics, the location identity of the entrance anonymous node can be easily revealed.

In cloud service, if an agent finds n suspicious files in the client's machine, it will send n virus-scan packets to the cloud. Attackers can generate the probing traffic based on this scenario. That is, by purposedly accessing a set of malicious files on the desktop, the agent is provoked to send out a sequence of requests to the cloud servers. If attackers customize the malware inter-accessing time, the corresponding virus-scan requests are considered as the probing traffic. Then, they can infer the next hop being

DOI: 10.1002/sec

used to relay these packets by analyzing the packet timing or volume signatures. This will continue until the whole circuit between the agent and the cloud server is discovered. Figure 10 shows the attack process. Attackers may not launch DDoS on the remote cloud servers but on the intermediate nodes instead, thus deteriorating QoS. As a result, the original requests will time out and will be resent, thus exacerbating the situation.

4.3. Distributed denial of service and economic denial of sustainability

The DDoS attack is still one of the most serious problems that needs to be tackled. Previous work in [5] shows that with more relay nodes compromised, anonymity systems under a selective denial of service attack become much more vulnerable than conventional security analysis would suggest. That is, packets relayed in a network with a majority of compromised nodes can be deanonymized more easily.

For AV in-the-cloud, DDoS can also negatively affect QoA by significantly delaying virus-scan-request packets in traversing the anonymous communication network. To protect cloud server clusters from DDoS, efficient DDoS detection and mitigation solutions have been offered by Internet service providers (ISPs) or security vendors, such as PeakFlow (www.arbornetworks.com) and Cisco Guard (www.cisco.com). On the other hand, currently, cloud data centers are built on virtualization technologies across the world. Scalable virtual imaging technologies are low cost by mounting new server virtual images to replace old ones corrupted by attacks.

Instead of launching large-scale DDoS attacks, a recent new counterpart, called economic denial of sustainability (EDoS) [9], has emerged. Nowadays, some vendors pay ISPs by traffic volumes or bandwidths. By controlling some desktop machines or using botnets, attackers can deteriorate QoS of the cloud network by generating probing traffic disguised as legitimate requests and by selectively affecting the reliability of a few anonymous nodes. Owing to the "weak-entrance-node" problem, such attacks can be easily staged. Instead of driving users away from the AV cloud systems, EDoS make these systems less reliable, though still functional. As a result, some customers may naturally attempt the communication again after the timeouts, resulting in more traffic congestions. By initiating stealthy attacks, attackers can subtly increase the traffic loads without triggering DDoS protection thresholds [10,11]. As a result, the whole cloud networking is still seemingly fine. However, EDoS attacks are eroding the profits because the security software companies, not the customers, pay for the bandwidth for both legitimate and disguised traffics.

4.4. Countermeasures and discussion

A countermeasure against traffic analysis attacks is to mix the cover traffic and the real traffic so that the total traffic in the links looks independent from the payloads. In our AV desktop solution, the agent can reshape the traffic patterns in all the links to the cloud servers so that every link presents a constant or similar traffic profile to attackers. The cover traffic may not be generated all the time but based on the incoming traffic statistical features; this is more efficient than inserting cover traffic at random. The randomness can often be removed by using statistical averaging methods. We have also embedded a request blocking filter inside the agent. Based on the reputation score calculated from the blacklisting and whitelisting, the filter can block the attacker's abusive requests.

Security standardization has not addressed the cloud yet; standards need to be made. For example, currently, there exist two kinds of anonymity networks: volunteer-based and commercial networks. The whole infrastructure is maintained by volunteers all over the world. Commercial companies can either build their anonymous systems by themselves or pay ISPs to maintain the systems. If something goes wrong with the location-hidden service, who will take the responsibility, ISPs or the cloud computing service providers?

To overcome the "weak-entrance-node" vulnerability, agreements regarding QoS, QoA, and SLAs should be reached between the customers and the vendors. Based on the operational models of the customers, in most cases,

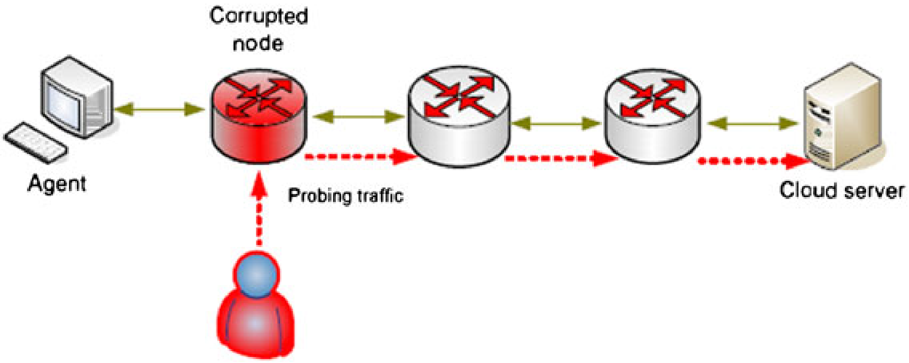


Figure 10. Weak-entrance-node attack.

what kind of specific service level should cloud service providers guarantee? On the customer side, the local network configurations must pass the penetration testing requirements before connecting to the cloud. A secure and robust desktop environment with low possibility of being compromised will reduce the abusive traffic and actualize economical saving for the providers.

5. CONCLUSION

Anti-virus in-the-cloud service is becoming the next-generation security infrastructure designed to mitigate virus threats. The service uses the anonymous network to hide identities of cloud servers. However, it is still vulnerable to traffic analysis attacks. In this paper, challenges and potential solutions in safeguarding AV in-the-cloud have been discussed. On the other hand, as more incoming malware samples become available, a powerful threat response system is required by AV in-the-cloud to support proactive detection and protection. SiC, an automatic signature generation scheme for zero-day malwares, has been proposed. Our approach is generic, and the test results have validated the ability and performances of SiC. We are still in the early stages, and several major issues in protecting AV cloud service remain to be addressed.

REFERENCES

1. Yan W, Zhang Z, Ansari N. Revealing packed malware. *IEEE Security and Privacy* 2008; **6**(5): 65–69.
2. Dingledine R, Mathewson N, Syverson P. Tor: the second-generation onion router. In Proceedings of the 13th USENIX Security Symposium, San Diego, CA, 9–13 August 2004.
3. Murdoch S, Danezis G. Low-cost traffic analysis of Tor. In Proceedings of the 2005 IEEE Symposium on Security and Privacy, Oakland, CA, 8–11 May 2005.
4. Bauer K, McCoy D, Grunwald D, Kohno T, Sicker D. Low-resource routing attacks against Tor. In Proceedings of the 2007 ACM workshop on Privacy in electronic society, Chicago, IL, 17–21 October, 2007.
5. Borisov N, Danezis G, Mittal P, Tabriz P. Denial of service or denial of security? In Proceedings of the 14th ACM Conference on Computer and Communications Security, Alexandria, VA, 29 October–2 November 2007.
6. Pietrek M. Peering inside the PE: a tour of the Win32 portable executable file format. *Microsoft Systems Journal* 1994; 15–34.
7. Yan W, Arrott A. Volume of threat: the AV update deployment bottleneck. In *Proceedings of the 19th Virus Bulletin International Conference*, Geneva, 23–25 September 2009.
8. Danezis G, Serjantov A. Statistical disclosure or intersection attacks on anonymity systems. In *Proceedings of 6th Information Hiding Workshop*, Toronto, 23–25 May 2004.
9. Hoff C. Cloud computing security: from DDoS (Distributed Denial of Service) to EDoS (Economic Denial of Sustainability). http://rationalsecurity.typepad.com/blog/2008/11/cloud-computing-security-from-ddos-distributed-denial-of-service-to-edos-economic-denial-of-sustaina.html [accessed November 2008].
10. Shevtekar A, Ansari N. A routed-based technique to mitigate reduction of quality of service (RoQ) attacks. *Computer Networks* 2008; **52**(5): 957–970.
11. Shevtekar A, Anantharam K, Ansari N. Low rate TCP denial-of-service attack detection at edge routers. *IEEE Communications Letters* 2005; **9**(4): 363–365.

DOI: 10.1002/sec