Burning money with firewalls

Article *in* South African Computer Journal · July 2015

DOI: 10.18489/sacj.v56i1.322

CITATIONS
0

READS
93

2 authors:

Ralf C. Staudemeyer
University of Applied Sciences Schmalkalden
34 PUBLICATIONS **414** CITATIONS
SEE PROFILE

James Connan
Rhodes University
56 PUBLICATIONS **225** CITATIONS
SEE PROFILE

Burning money with firewalls

Ralf C. Staudemeyer*, James Connan†

*School of Computing, University of South Africa, Johannesburg, South Africa
†Department of Computer Science, Rhodes University, Grahamstown, South Africa

1 INTRODUCTION

Protecting computing infrastructure is an expensive but necessary task. Companies that run their own computing infrastructure need to be constantly vigilant against intruders on their networks and groups or individuals that seek to intercept their communications.

Companies turn to firewall and intrusion detection system vendors to help them safeguard their information and infrastructure. These companies are putting their faith in expensive devices that attempt to distinguish between friendly and malicious traffic.

Despite the expense and complexity of these systems, malicious users seem to find ways to avoid detection and news of massive breaches have become an almost daily occurrence. We propose to fundamentally change the way to look at information security. We highlight some of the fundamental flaws in current systems and expose risks that many experts know, but do not adequately take into consideration.

2 FIREWALLS AND INTRUSION DETECTION SYSTEMS

Firewalls and intrusion detection systems (IDS) are used to protect computing infrastructure. While both systems endeavour to prevent malicious access to computing infrastructure, they do so using very different approaches.

Firewalls track the traffic between attached networks in order to protect machines from unwanted access. This is done by user configured rules, traditionally based on sender, recipient and the service used. Firewalling is a mature technology that has been through multiple evolutions. First-generation firewalls were packet filter (PF) devices limited to observing single IP packets. Second-generation devices were enhanced to track and filter TCP (and later, UDP) network connections; these are termed stateful packet inspection (SPI) firewalls. Third-generation systems, termed deep packet inspection (DPI) devices or application layer firewalls, are able to identify and track services on the application layer by using protocol signatures. Recent devices can perform this task in hardware at near switching speed [1].

Email: Ralf C. Staudemeyer `ralf@staudemeyer.de`, James Connan `j.connan@ru.ac.za`

Intrusion detection systems, in contrast to Firewalls, are concerned with the (automated) detection of manual or automated attacks on computer systems [2]. This is done by monitoring and analysing events associated with network traffic [3]. Since intrusion detection is a computationally intensive task, most systems work with a time delay to avoid interfering with user activities. There are two types of IDS:

- Systems that analyse the local data of computer systems in order to detect attacks on that specific host, are termed Host Intrusion Detection Systems (HIDS).
- Systems that run on specialised network equipment, collecting network data, are termed Network Intrusion Detection Systems (NIDS). Here, network data is recorded and analysed, partially manually by an expert, in an attempt to identify policy violations or other inappropriate use of the system deeply buried in network traffic.

Well-known examples of successful open-source IDSs are Bro[1] [4], Snort[2] [5], Tripwire[3] [6], Samhain[4] [7] and Prelude[5] [8].

In recent years, Intrusion Prevention Systems (IPS) have emerged. These are being merged into traditional firewall systems as an additional feature. IDS and firewalls are increasingly converging into Unified Threat Management (UTM) devices and will in all likelihood merge into one single device.

To summarise: PFs are easy to implement, but the need to inspect every packet means that they do not scale well in terms of the processing power required. This shortcoming was addressed by SPI, at the cost of being much more memory-intensive; however, once the memory of an SPI system is exhausted, it is easily exploitable. DPI introduced a much more complex set of network usage policies at the cost of being computationally very expensive.

The rules provided by PFs are simplistic in nature and rely on IP header information to make decisions. SPI overcame this limitation by keeping track of connections. The latest generation of firewalls, supporting DPI, addressed most of these concerns, but are very expensive to maintain terms of computational and human resources required.

[1] www.bro.org
[2] www.snort.org
[3] sourceforge.net/projects/tripwire
[4] www.la-samhna.de/samhain/
[5] www.prelude-ids.org

3 PROTECTING INFORMATION

As the internet becomes ever more ubiquitous and more and more personal data finds its way onto the internet, data can be protected by encrypting information on transport and application layer, and by implementing means to resist traffic analysis.

In recent years encrypting traffic has become more common. It is no longer just business transactions that are protected by encryption, but the constant fear of identity theft and privacy concerns around the tracking of individuals have forced more and more online services to encrypt traffic. Encryption used to be limited to the domain of e-commerce, banking and business traffic, but now companies like Facebook and Google also offer encrypted channels for communication with their servers.

Unless a network is extremely limited in its use, it is unlikely that encrypted traffic can be banned from the network. Firewalls were not designed with encrypted traffic in mind. Hence, firewalls are unable to successfully manage legitimate or malicious encrypted traffic. Systems like ssl-proxies attempt to address this shortcoming, but it has many disadvantages. From a user perspective it defeats the very purpose of encrypted traffic.

In constrast network traffic analysis focuses on the metadata and ignores content information. This includes endpoint addresses, timing and location information. Traffic analysis can be addressed by anonymity systems.

Anonymising proxy networks have existed for a long time and started with the implementation of Chaum's Mix in 1981 [9]. A Mix network tunnels encrypted traffic through a number of low-latency proxies.

Initially interest in this field was primarily theoretical but in the last 30 years a lot of research in this field has looked at developing practical and usable systems for preserving anonymity [10, 11]. Such systems include Mixmaster [12], Mixminion [13], TOR [14], I2P [15] and many others which are actively under development.

The negative effects of anonymity, such as the obfuscation of criminal activity, child pornography and abusive behaviour is often highlighted. However, anonymity also makes it harder for oppressive regimes to suppress freedom of speech, protecting against bias such as status, gender and race, and allowing freedom of expression without fear of personal repercussions. It is on us citizens to decide how much we value these positive properties against the others.

4 WRAP-UP

Historically firewalls and Intrusion Detection Systems were built to protect network services and to detect intrusions. Today, these systems are unable to deal with end-to-end secured traffic. This renders them useless on successfully secured connections that use end-to-end encryption and implement the anonymity property.

In terms of end-to-end security, encryption works well to protect content data, but does not address leakage of metadata information. Metadata can be used to build up knowledge about the communicating parties as well as the content of the communications. It provides access to all sorts of information, including leaking detailed content information without breaking encryption. Exactly what information can be extracted is unpredictable.

The anonymity property provides protection against traffic analysis. It is to information leakage what encryption is to data protection. By using anonymity systems like TOR and I2P, there are effective ways to communicate while both encrypting data and minimising information leakage by preserving anonymity.

The inability to deal with encrypted data and anonymised connections mean that firewalls, IDS and UTM are unlikely to be a sound long-term solution for protecting computing infrastructure. From the viewpoint of data protection these devices frequently fail since attackers successfully hide their activities. Even worse, these systems actually force users to not protect transported data and communication channels to support content analysis.

What has been successfully accomplished is to build feature rich surveillance systems that help to enforce complex policies and depend on highly skilled operators.

Those who wish to protect not only their servers, but actually confidential information, will have to rely more on people then on devices to protect their systems and will need to look at ways of reducing their information leakage. Therefore we conclude that data should be secured by the communication partners with a best-effort approach. Encryption protects against direct access of data, given a wise choice of the encryption algorithms, nested encryption and strong random number generation. An additional layer of security is needed to protect against traffic analyses. This can be provided by anonymity systems.

We call for a paradigm change, away from reliance on security devices that enforce lower security, are challenging to operate, implement surveillance, and increasingly fail to protect sensitive information. We strongly recommend considering transport as unsecure by definition and actively protecting information from any potential eavesdropping party.

We are confident that classical FW, IDS and UTM security devices will fail. Futhermore security policies are of little use since it is unclear what information can actually be extracted from available data.

To fully protect confidential information it is necessary to ensure it is not saved on a computer. In many use cases this has become unrealistic. Encryption and anonymity are both achievable with reasonable effort. We strongly recommend using both.

REFERENCES

[1] Palo Alto Networks. "Palo Alto Networks", 2013. URL `http://www.paloaltonetworks.com/`.

[2] S. Axelsson. "Intrusion detection systems: A survey and taxonomy". *ACM transactions on information and system security (TISSEC)*, vol. 3, no. 3, pp. 15–99, 1999.

[3] K. Scarfone and P. Mell. "Guide to intrusion detection and prevention systems (IDPS)". Tech. Rep. 2007, NIST: National Institute of Standards and Technology, U.S. Department of Commerce, 2007.

[4] V. Paxson. "Bro: A system for detecting network intruders in real-time". *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[5] M. Roesch. "Snort–Lightweight intrusion detection for networks". *Proceedings of the 13th USENIX conference on system administration*, pp. 229–238, 1999.

[6] G. H. Kim and E. H. Spafford. "The design and implementation of Tripwire: A file system integrity checker". *Proceedings of the 2nd ACM conference on computer and communications security*, pp. 1–18, 1994.

[7] B. Wotring, B. Potter and M. J. Ranum. *Host integrity monitoring using Osiris and Samhain*. Syngress Media Inc, 2005.

[8] Prelude. "PreludeIDS". World Wide Web electronic publication, 2011. URL `http://prelude-ids.org/`.

[9] D. L. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms". *Communications of the SACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.

[10] G. Danezis and R. Clayton. "Introducing traffic analysis". In *Digital privacy: Theory, technologies, and practices*, pp. 1–24. CRC Press, 2007.

[11] A. Ruiz-Martínez. "A survey on solutions and main free tools for privacy enhancing Web communications". *Journal of network and computer applications*, vol. 35, no. 5, pp. 1473–1492, Sep. 2012.

[12] U. Möller, , L. Cottrell, Anonymizer, Inc., P. Palfrader, the Mixmaster Project, L. Sassaman and Nomen Abditum Services. "Mixmaster protocol, version 2 (draft)". Tech. Rep. C, 2003. URL `https://tools.ietf.org/html/draft-sassaman-mixmaster-00`.

[13] G. Danezis, R. Dingledine and N. Mathewson. "Mixminion: Design of a type III anonymous remailer protocol". In *Proceedings of the 19th international conference on data engineering*. 2003.

[14] R. Dingledine, N. Mathewson and P. Syverson. "Tor: The second-generation onion router". In *Proceedings of the 13th USENIX security symposium*, vol. 13, pp. 303–320. USENIX Association, 2004.

[15] Zzz and L. Schimmer. "Peer profiling and selection in the I2P anonymous network". In *Proceedings of the fourth privacy enhacing technologies convention PET-CON 2009*, p. 20. 2009.