Virtualized Dynamic Port Assignment and Windowed Whitelisting for Securing Infrastructure Servers

Ronald Loui, Lucinda Caughey,
Mohammad Ghasemisharif
Dept. of Computer Science
University of Illinois
Springfield, IL
rloui2@uis.edu

Rogelio Salvador
Information Technology Services
University of Illinois
Springfield, IL

***Abstract*—We describe a novel method of securing services by adding windowed whitelisting to an arbitrary and constantly changing assignment of services to ports (or virtual ports). This is aimed at mitigating port scanning threats and unauthorized intrusion attempts, and to protect a community of known users from data loss. In essence, port numbers, time, and IP address will be used as part of the password/access mechanism; this segregates traffic so that content-based restrictions can be more effective. It also provides a connection-based security wrapper for services that might be vulnerable to software exploits, such as the buffer overruns and backdoors.**

The method requires a portal to authenticate users and disseminate knowledge of the current port assignment, in addition to permitting users to request a "window" of time to be white-listed. It requires a firewall with dynamic port and whitelist reconfigurability. The method is intended to enhance byte frequency histogram analysis and regexp restriction of traffic. It also requires a policy for keeping alive long-lasting connections. It can be implemented easily with virtual ports using redirection. We discuss some implications for web page rewriting and cgi security, as well as legacy services such as ssh and sftp. The effect is to create a cross-product of IP range, port range, and time specificity, to create a large and sparse search space for any adversary.

I. Introduction

Systems administrators constantly face the decision to open a service on a port, or not, when configuring servers. Although the careful sysadmin is reluctant to provide too many services on open ports, interactive users on mixed-use servers and even dedicated single-use servers often require opening ports periodically.

One aspect of the security problem introduced by opening ports is the lack of control over authentication at the port, due to the protocol associated with the service. Some popular legacy services are known to have vulnerabilities, such as telnet on port 23, or attract intruder attention, such as ssh on port 22. Other services require careful examination of server-side programming, such as cgi scripts on port 80. In many cases, custom authentication cannot be added on top of the standard access control once the port has been opened. Meanwhile, some strategies for intelligent firewall or data loss prevention use byte frequency histograms, regular expression filtering, or a simple list of permissible characters. When different kinds of traffic are mixed at a single port, such as all web traffic at port 80, these security methods are considerably less effective. Similarly, when multiple users with multiple purposes are all whitelisted at a single port (or worse, when multiple users are all whitelisted for all ports on a server), the address-based or id-based restriction is weakened.

Usually services are assigned to standard ports, with the consequence that they can be attacked without guesswork. Even when ports are assigned in a nonstandard way, as ssh is often assigned to a port other than 22, hackers will easily find which services are open at which ports with quick scanning. Once hackers find services on open ports, they can begin IP-spoofing to defeat address-based whitelisting. Without nuanced, time-dependent, finer-grained whitelisting, even a dynamic, randomly assigned port can be quickly discovered.

We propose assigning ports differently on servers, in a dynamic fashion, in conjunction with finer grained whitelisting of IP addresses, and with several refinements. This will create a larger search space for scanning and spoofing, especially because time becomes a third dimension in the search space. This permits firewalls to have greater impact on access to the server, and can even mitigate some DDOS attacks. Because users must register to discover services, modern, layered authentication protocols can be used as security wrappers for older protocols. And users can be asked to be specific about what form of data and how much data they will transfer.

By assumption, we have in mind those servers that do not need to open a large fraction of the available 65535 TCP ports, those that can start and stop services without rebooting (or that have virtual port forwarding), and those that can dynamically add or subtract to IP whitelists on a firewall). In fact, the focus here is on infrastructure servers, where security is more important than convenience, and infrequent legitimate access

978-1-4673-9985-2/16/$31.00 ©2016 IEEE

by a small number of users is the norm. For these servers, occasional maintenance is required by a small group of privileged users, which is anathema to the idea of opening a service on a port, protected only by a password, or even a password and a permanently whitelisted address. range

II. Main Idea

The main idea is to break the relation between service and port, and insert a web-based authentication, upon which the systems administrator can exercise greater control. We call it a dynamic port assignment with windowed whitelisting (DPA+WW) strategy.

1. Close all ports by default, except the one http/https port where users are authenticated and register for access.

2. At regular intervals, such as each day or every few hours, assign a new port, somewhat arbitrarily, for each service. For example, one day ssh will be on port 49155, and the next day it will be on 33435. Start each service on this new port, and stop service on any prior ports where there are no current connections. Avoid standard port number assignments.

3. Require all ports to be accessed through a whitelist that checks full IP addresses (not just prefixes).

4. Create a web service on a fixed open port, or a port that varies algorithmically by date and time, and possibly also by IP address, that reports which service is currently available on which port. This web service should require normal password or layered authentication, and this authentication is assumed effective. This is a PORT PUBLICATION service, which is one part of a CONNECTION MANAGER. Inform the community of potential users that this URL is used for the port publication web service.

5. Users join the whitelist on a port only through this service. This is a ticket to access the service, and a user is whitelisted only for a short window of time, which might not even always begin immediately. Users are removed from the whitelist for that port when their session terminates, or simply times out. Require users to specify meta-information about their session: time limit, byte-in limit, byte-out limit, special character limits, etc., which may be menu-driven to provide typical session characteristics.

6. Drop any traffic on an unused port; drop any traffic on a port from an IP address not whitelisted, except for traffic on the CONNECTION MANAGER port.

III. Analysis

The main benefit of dynamic port reassignment is that whitelisting becomes 65000x more effective against IP-spoofing, because attackers must iterate through port numbers AND IP addresses simultaneously. Port scanning becomes less effective, depending on how many IP addresses are whitelisted and how many services are held open by long sessions or long timeouts, because attackers must iterate through the cross product of two large ranges. Each range is small enough to be sampled by a determined adversary, but the product of the ranges is probably too large, especially when modest delays are introduced. The port-x-IP-address pairs are sparse, but the port-x-IP-address-x-time-window triples are even sparser, perhaps by several more orders of magnitude. Without our method, the attacker can project the three-dimensional space down to a single dimension, and search that dimension exhaustively.

This improvement in security is based on enlarging the search space and reducing the attack surface. We recommended it here as a defense-in-depth measure based on apparent randomization. The enablement of fine-grained whitelisting (full IP restriction, rather than prefix-restriction, and time-window-whitelisted rather than persistently-whitelisted) is a significant advantage.

One could open extremely vulnerable legacy services such as ftp and telnet under this strategy, but it is not advised. Of course, if one had to open such a vulnerable service, this dynamic port assignment-with-windowed-whitelisting (DPA+WW) strategy may afford some temporary protection. The reality even for the more secure services is that vulnerabilities are constantly being discovered, patching takes time, and open ports attract malicious traffic (for example, ssh attracts password guessing intrusion attempts which are a nuisance even if they are unsuccessful). The recent discovery of a glibc vulnerability affecting ssh and sudo is an example of such unknown vulnerability.

Also, by assigning no services to their standard ports, firewalls can block any traffic based just on the port number, for the most common port numbers, which may constitute the bulk of nuisance traffic. In many cases, this will significantly reduce exposure to DDOS attacks, especially from script kiddies. Using non-standard ports may have implications on a user's outgoing software and firewall, which might hard-code standard ports for services on a target server.

The CONNECTION MANAGER must have a component that determines when to remove IP-addresses from the port's whitelist. One way to enforce connections not to be held open too long is simply to disconnect after a time (a hard window). A better way, when practicable, is to negotiate the window during the authentication step in the PORT PUBLICATION web service. In this way, the user's estimates of connection time contribute to the security of the server. Both can be used in conjunction with an idle-time hang up.

Sometimes a service is accessed automatically and/or routinely, and the port must be hard coded in a script (for example, a daily automatic database feed). In a DPA+WW scheme, the username and password, and possibly the anticipated window of time, would be written into the script

instead of determined via a webpage. If ports are assigned according to algorithm, or by rotation through a list, a script could hard code a port-attempt sequence; however, registration of the IP for windowed whitelisting should still be required rather than accepting traffic from such an address as persistently (or "permanently") whitelisted. This would be a one-way notification, so it could still simplify the scripting.

Sometimes an unknown user with an unanticipated IP-address will want to use a service de novo, and the sysadmin would like to support that use. Most such cases will be http traffic on port 80 or https on port 443, though one can imagine SMTP/SMTPS mail traffic on port 25 and 465, or DNS on port 53, or MySQL on port 3306, from new, external users. In the case of web traffic, port reassignment can be managed by the application that generates outgoing URLs (see below). Other initial contacts , such as mail, may not be able to benefit from this DPA+WW scheme because they cannot insert a custom authentication stage ahead of a protocol with redirection.

Because of the potentially large insider community that can use the PORT PUBLICATION service, this method remains vulnerable to insider threats. Also, packet sniffers will be able to reveal port and IP address combinations quickly enough to exploit time windows. Our method is mainly aimed at preventing external users from exploiting well known or recently discovered security holes in standard services. Our targets are external intrusion attempts (including their nuisance traffic), and external denial of service attempts. Even when there are insider threats, though, firewall pattern matching improved by traffic segregation should reduce data loss to malicious insiders.

Assuming that a service will be in use 1% of the time, 99% of malicious traffic occurs on standard ports, pattern-based BFH or REGEXP filtering is 99% effective, and 10 users sharing a service at a time on the same port, various combinations of the components can be compared, using rough calculations:

Standard Static Port Assigned (SSPA), no user-IP filtering, no Segregation, no Content restriction (normal server):

can be attacked by scripts aimed at known ports

Non-Standard Assigned Static Port (NSPA), no user-IP filtering, no Segregation, no Content restriction ("use port 23 for ssh"):

can be port scanned, then attacked after discovering ports, 1/65k

can drop traffic at standard ports, 1/100

Random Dynamic Port Assignment (DPA), no user-IP filtering, no Segregation, no Content restriction (port-hop):

can be port scanned, then attacked after discovering ports while port is in use, 1/65k * 1/100

can drop traffic at ports not in use, 1/100 * 1/100

Random Dynamic Port Assignment (DPA), user-IP prefix filtering, no Segregation, no Content restriction (port-hop with firewall):

can be port scanned with whitelisted user-IP-prefix guessing (which is usually easily guessed), then attacked after IP x port combination discovered, while port is in use; either ports and users are 1-to-1 which places a limit on number of users, or rate of change is restricted, min(1/256* 1/256, 1/65k * 1/100 + 1/256)

can drop traffic at ports not in use with user-IP prefix filtering, 1/100 * 1/100 * 1/100

Random Dynamic Port Assignment (DPA), user-IP full-address filtering (WW), some Segregation, no Content restriction (DPA+WW):

can be port scanned with IP spoofing while port is in use, 1/65k * 1/100 * 1/256 * 1/256 * 1/256 * 1/256 * 10

can drop traffic at ports not in use with user-IP full filtering, 1/100 * 1/100 * 1/256 * 1/256 * 1/256 * 1/256 * 10

Random Dynamic Port Assignment (DPA), user-IP full-address filtering (WW), some Segregation, some Content restriction (DPA+WW+BFH/REGEXP):

can be port scanned with IP spoofing while port is in use if probing payload matches restriction, 1/65k * 1/100 * 1/256 * 1/256 * 1/256 * 1/256 * 1/100 * 10

can drop traffic at ports not in use with user-IP full filtering unless content matches restriction, 1/100 * 1/100 * 1/256 * 1/256 * 1/256 * 1/256 * 1/100 * 10

IV. Data Exchanged Through Web Services

In the special case of web services on port 80, especially those that exchange user input or potentially significant output data, the new paradigm suggests assigning each page, or part, of the session dialogue to a different dynamically assigned port. In this way, firewall filtering can check for injection and buffer overrun attacks specific to each page, and the dynamic ports implement an internal session id.

If a webserver faces a legitimate user community through a small set of dynamic pages, those pages can each be considered a separate service, and assigned dynamic ports independently.

In a banking dialogue, for example, the home page could be on port 80, but the page where the username is input could be dynamically assigned to port 51001 (in this cycle of reassignment), and the subsequent page where the password is input could be dynamically assigned to port 9804 (in this cycle of reassignment). Meanwhile, the bank balance report could redirect to port 10222 (in this cycle of reassignment). Not only would the dialogue at each stage be whitelisted for the expected user in the expected time window, but data to and from the server could be inspected with tighter constraints on validity (see the next section). This will prevent certain kinds of session hijacking, and make data loss prevention more specific to the expected pattern of output on each page.

The DPA+WW strategy can be implemented easily with a URL-parser and port-rewriting script for every HTML page served. From the content management point of view, this is a transparent sysadmin function. To the user, the only effect, which is not a trivial effect, is that web page links grow stale after a short time, when ports are reassigned. This is not necessarily a bad thing for interactive data-exchanging systems that have limited login times for other security reasons.

Although the page may have dozens, even hundreds of links, most static pages will be directed through the same port. It is only the pages that can exchange important user data with the server that are desired to be segregated.

V. Byte Frequency Histogram and Regexp Firewall Enhancements

A common strategy for protecting servers is to restrict the set of characters, or string patterns, that may pass into or out of a port. Advanced firewalls are capable of regular expression pattern matching, and analysis of byte frequency histograms.

One restriction might be to reject input that contains null characters, which are commonly used for buffer overflows in c programs. Another restriction might be to block payloads containing short virus signatures at the firewall. The problem with these restrictions is that the more varied the input on a port, the less constrained must be the string-pattern or character restriction. In a sense, merging traffic at a port forces content restriction to the lowest common denominator. If the buffer overrun prevents, for example, images to be uploaded to a server using the same port, then the restriction is usually removed.

On the other hand, when a server is expecting just one kind of user input, the restriction can be very specific. Much data loss can be prevented by describing expected output as a regular expression and placing a limit on expected length. URI-translation may be unnecessary when special characters are excluded (for example, semi-colons, double quotes, question marks, and other characters used to inject executable code) – the firewall can drop the attempted communication. The method is to segregate traffic at the intra-dialogue level. For dedicated servers that do not merge many different kinds of web services, this should be easy. But even when there is diverse and shared content on a single server, expected input and output types can be grouped using a few well chosen pattern “classes” (passwords, numeric data, encrypted BFH-balanced binaries).

Improved intelligent firewall protection through segregation of traffic may seem like an after-thought. But segregating cgi and ajax traffic at the page-level, to make finer filtering and pre-processing possible, was actually the genesis of the idea to do fine-grained port reassignment.

VI. Using Virtual Port Translation

We have so far been contemplating a reconfiguration of the server so that ports do not have standard uses, and constant communication with a firewall is required to maintain address-based access control. However, it is possible to leave the server in its normal state, or in a slightly perturbed but static state, and do all of the work at the firewall through port translation. The firewall would implement virtual ports, which would undergo dynamic reassignment cycling, but the server itself would simply be the beneficiary of the translation.

The PORT PUBLICATION and authentication/registration would still be handled by a normal server (not necessarily the server with dynamic ports). CONNECTION MANAGEMENT of whitelists and timeouts, and possibly output URL rewriting and regexp pattern matching, would also not normally be implemented in dedicated firewall hardware. But they could be implemented in a general purpose machine assigned as a firewall.

Virtualizing ports may sometimes mean that more than a 16-bit port number can be used. In most situations, software and address propagation will not support the use of long port numbers, but if a larger range of ports were to be supported in transport, virtualization could make use of this even if servers maintain a more limited range.

IP addresses could also be used in sparse target port redirection. If a single machine could use all of the suffixes from 0 to 255, it could assign suffixes with windowed whitelisting and dynamic reassignment. For example, if the server is actually using 192.102.230.101, and 192.102.230.100 is unused, then the CONNECTION MANAGER could temporarily assign ssh traffic to 192.102.230.100:1167 and whitelist new connections there. An adversary would have to very quickly guess the whitelisted IP, the port, and the suffix.

We tested dynamic port assignment using pfSense for forwarding, and ufw for temporary whitelisting. A simple awk script is used to implement the CONNECTION MANAGER, for cgi requests for service, and for removing whitelisted services, ports, and IP addresses from ufw after timeout.

Virtual port forwarding is our preferred embodiment because it does not require restarting services on different ports, and it permits multiple ports for services that are normally configured to run on one port at a time such as ssh.

VII. Related Work

Dynamic port assignment has been discussed in relation to load balancing for improved performance, e.g., Anderson (1989) for terminal devices, Fernandez (1990) for modems, Zornig (1998) for hubs, Blumenau (2001) to reduce blocking, Lu (2006) for private network address translation, Elo (2007) for connection speed management. There seems no prior thought on reassignment for security, because servers are assumed to offer services on standard ports (how else would users know where to go?).

Our idea adds the connection step (as does Morris), and crucially, adds windowed whitelisting.

A different related work is called "port hopping" by Brand (2004) which is based on calculated reassignment of port, which were followed by Lee (2004), Badishi (2005) RPH protocol, Shi (2007) also based on calculated synchronization, and Morris (2012) which changes ports within a sender-receiver dialogue. Nice (2013) is a recent summary of work in this area.

The main difference in our work is that we use port reassignment with windowed whitelisting not just to hide the service, but to permit traffic segregation for better automated analysis and filtering. Thus, the port-hopping protocols focus on the HOPPING, typically an algorithmic or shared-cryptographic hop, while our work focuses on the SEGREGATION and SPARSITY of ASSIGNMENT, which improves the effectiveness of related intrusion and data loss prevention techniques. For example, Larsen (2011) in IETF best practices focuses on the selection of the randomized port, with no mention of whitelisting. Port hopping is aimed at the synchronization problem when resource availability is dynamic. Our work is aimed at server management to secure an internal community from external threats, especially for protecting static resources. We solve the synchronization problem simply by delegating authentication to a standard web-based preliminary stage. Ours is a wrapper, especially useful at adding modern security mechanisms to older protocols, while theirs is a kind of exchange switching.

Regular expression filtering of traffic at the firewall was discussed by Lockwood (2006) and Moscola (2003). Statistical analysis using byte frequencies is discussed in Provos (2003), Wang (2004), Li (2005), Tabish (2009), etc.

There are of course many protocols that require a registration stage before users may access resources at a hidden and/or protected address. These are usually implemented using session IDs or statically assigned addresses and are not thought appropriate for services assigned to ports on servers. Some of these protocols do segregate the dialogue into distinct parts on different ports, for example "active FTP" (Slacksite.com).

VIII. Conclusion

At first, this may seem like a familiar story: combine methods so that the multiple is more effective than each component. That is part of the picture: windowed whitelisting makes dynamic port assignment more effective, and dynamic port assignment makes windowed whitelisting more refined. Compared to port hopping, which should be more widely adopted in our opinion, the effectiveness is a true multiple rather than two separate serial minimizations. Perhaps with the effectiveness of DPA+WW over a random port hopping protocol will result in wider use of dynamic ports.

But our initial motivation was to make intelligent firewalls more effective, by permitting classification and filtering to occur on finer grained classes of traffic, separated by purpose. One way to achieve this was to segregate traffic using more ports. Nonstandard, dynamic, multiassignment of ports was a consequence. We started wanting to protect cgi by separating pages onto different ports, and ended up controlling access on all ports. In our bigger picture, merging of traffic is bad, because it makes policing that traffic harder. Merging purposes, merging users, and merging time are all responsible for the dilution of security.

In the real world, as opposed to the world of computer communications, people who want to connect with a bit of privacy agree on undisclosed, arbitrary, and temporary locations, and their choices of location match their transactional requirements. This is dynamic port assignment and windowed whitelisting. It seems foolish not to use these same ideas for the secure management of computer servers.

References

[1] Atighetchi, Michael, et al. "Adaptive use of networkcentric mechanisms in cyber-defense." Object-Oriented Real-Time Distributed Computing, 2003. Sixth IEEE International Symposium on. IEEE, 2003.

[2] Atighetchi, Michael, et al. "Adaptive cyberdefense for survival and intrusion tolerance." Internet Computing, IEEE 8.6 (2004): 25-33.

[3] Anderson, Gary D., et al. "Dynamic port reconfiguration." U.S. Patent No. 4,868,783. 19 Sep. 1989.

[4] Badishi, Gal, Amir Herzberg, and Idit Keidar. "Keeping denial-of-service attackers in the dark." Distributed Computing. Springer Berlin Heidelberg, 2005. 18-32.

[5] Brand, Thomas, et al. "Denial of service protection through port hopping." U.S. Patent Application 10/951,466, 2004.

[6] Dainotti, Alberto, et al. "Identification of traffic flows hiding behind TCP port 80." Communications (ICC), 2010 IEEE International Conference on. IEEE, 2010.

[7] De Vivo, Marco, et al. "A review of port scanning techniques." ACM SIGCOMM Computer Communication Review 29.2 (1999): 41-48.

[8] Ehrenkranz, Toby, and Jun Li. "On the state of IP spoofing defense." ACM Transactions on Internet Technology (TOIT) 9.2 (2009): 6.

[9] Elo, Anders, Andreas Öman, and Magnus Lundström. "Dynamic port configuration of network equipment." U.S. Patent No. 7,174,371. 6 Feb. 2007.

[10] Fernandez, Raul F., and Ignacio Urbieta Jr. "Inband dynamic port allocation." U.S. Patent No. 4,893,305. 9 Jan. 1990.

[11] Guenthner, Timothy John, et al. "Mapping web server objects to TCP/IP ports." U.S. Patent No. 6,360,262. 19 Mar. 2002.

[12] Labovitz, Craig, et al. "Internet inter-domain traffic." ACM SIGCOMM Computer Communication Review 41.4 (2011): 75-86.

[13] Larsen, M. and F. Gont, Recommendations for Transport-Protocol Port Randomization, Best Practice, Internet Engineering Task Force (IETF) RFC 6056, January 2011. ISSN: 2070-1721, January 2011

[14] Lee, Henry CJ, and Vrizlynn LL Thing. "Port hopping for resilient networks." Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th. Vol. 5. IEEE, 2004.

[15] Li, Wei-Jen, et al. "Fileprints: Identifying file types by n-gram analysis." Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC. IEEE, 2005.

[16] Lockwood, John, et al. "Methods, systems, and devices using reprogrammable hardware for high-speed processing of streaming data to find a redefinable pattern and respond thereto." U.S. Patent No. 7,093,023. 15 Aug. 2006.

[17] Lu, Wei, and Junan Duan. "Dynamic port management." U.S. Patent No. 6,983,319. 3 Jan. 2006.

[18] Morris, Cameron Craig, Lloyd Leon Burch, and David Thomas Robinson. "Techniques for port hopping." U.S. Patent No. 8,301,789. 30 Oct. 2012.

[19] Moscola, James, et al. "Implementation of a content scanning module for an internet firewall." 11th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM 2003.

[20] Nice, T., and A. Mathew. "Different Types of Port-hopping Methods Used to Prevent DDoS Attacks." International Journal of Computer Science Research & Technology 1.5 (2013): 36-37.

[21] Provos, Niels, and Peter Honeyman. "Hide and seek: An introduction to steganography." Security & Privacy, IEEE 1.3 (2003): 32-44.

[22] Shi, Leyi, et al. "Port and address hopping for active cyber-defense." Intelligence and Security Informatics. Springer Berlin Heidelberg, 2007. 295-300.

[23] Slacksite.com, "Active FTP vs. Passive FTP, a Definitive Explanation," http://slacksite.com/other/ftp.html.

[24] Song, Jae Min, Byung Jae Park, and Hee Tae Yoon. "System and method for transmitting and receiving session initiation protocol messages." U.S. Patent Application 14/003,161.

[25] Tabish, S. Momina, M. Zubair Shafiq, and Muddassar Farooq. "Malware detection using statistical analysis of byte-level file content." Proceedings of the ACM SIGKDD Workshop on CyberSecurity and Intelligence Informatics. ACM, 2009.

[26] Wang, Ke, and Salvatore J. Stolfo. "Anomalous payload-based network intrusion detection." Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, 2004.

[27] Zornig, John G., Tavit K. Ohanian, and George A. Klarakis. "Load balancing port switching hub." U.S. Patent No. 5,742,587. 21 Apr. 1998.