Optimizing Internet Scanning for Assessing Industrial Systems Exposure

Jérôme François, Abdelkader Lahmadi, Valentin Giannini, Damien Cupif, Frédéric Beck, Bertrand Wallrich

Optimizing Internet Scanning for Assessing Industrial Systems Exposure

Jérôme François*, Abdelkader Lahmadi*†, Valentin Giannini‡,
Damien Cupif‡, Frederic Beck* and Bertrand Wallrich*

*Inria Nancy Grand Est, Nancy France, {jerome.francois,frederic.beck,bertrand.wallrich}@inria.fr
†LORIA - Université de Lorraine, Nancy, France, abdelkader.lahmadi@inria.fr
‡Telecom Nancy, Université de Lorraine, Nancy, France, {valentin.giannini,damien.cupif}@telecomnancy.net

***Abstract*—Industrial systems are composed of multiple components whose security has not been addressed for a while. Even if recent propositions target to improve it, they are still often exposed to vulnerabilities, since their components are hard to update or replace. In parallel, they tend to be more and more exposed in the public Internet for convenience. Although awareness of such a problem has been raised, there is no precise evaluation of such a risk. In this paper, we define a methodology to measure the exposure of industrial systems through Internet. In particular, a carefully designed scanning approach, named WiScan, is proposed with a low footprint due to the high sensitivity and low resources of targeted systems. It has been applied on the entire IPv4 address space, by targeting specific SCADA ports.**

***Index Terms*—Internet scanning, IPv4 scanning, SCADA, Industrial systems, security, assessment**

I. Introduction

Cyber-Physical Systems (CPSs) become popular with the advent of the Internet-of-Things, but Industrial Control Systems (ICS), as for example SCADA (Supervisory Control and Data Acquisition) systems, have been deployed for a long time. Those systems are used in many environments: hospitals, factories, nuclear power plants, etc. Thus, they are frequently part of critical infrastructures, and a failure can be dramatic with human lives being endangered [1]. Besides, they tend to be more and more connected to Internet for many reasons: integration with standard information systems for a better management at all levels (commercial, technical, etc.), remote maintenance [2], even by third parties, etc. As a result, they are also more exposed to malicious actions. Recently, Stuxnet [3] has highlighted the real risk of such threats by targeting nuclear facilities, but the first attacks occurred in the 80's [4]. Because attacks against ICS are increasing [5], assessing how they are exposed in the Internet is of capital importance to explore the impact of attacks on them, and also to be able to propose defensive mechanisms.

Even if the security of such systems has been empowered [6], the National Institute of Standards and Technologies (NIST) still recommends a strong separation of the industrial network from the corporate network and the Internet [7]. This requires a multi-layer architecture, where field devices, *e.g.* PLCs (Programmable Logic Controllers), cannot be directly accessed, but only through dedicated servers, *e.g.* for programming or maintenance. Those latter have to be protected by access control mechanisms and firewalls. Therefore, field SCADA devices should not be visible from Internet. In this paper, we actually demonstrate that many of them are exposed, by scanning the entire IPv4 address space.

We designed WiScan, and used a low footprint scanning method, similar to what an attacker could leverage. Because ICS are not usual IT systems, in particular with usually less resources, they are more sensitive to unexpected loads. Hence, scanning these systems may have also unpredictable and fatal consequences, as for instance stopping a system. Even an attacker may not desire such a result from a scan, since this prevents him from further actions once he identifies some devices. Indeed, he would have preferred to access the system, in order to silently perform other actions (retrieve information, reconfigure the system to slowly degrade its functioning, access other devices, etc.).

To summarize, our contribution is three-fold:

- definition of a low footprint scanning methodology, WiScan, well suited for sensitive environments such as ICS,
- comparison of our methodology with recent state-of-the-art approaches: ZMap [8] and Masscan [9],
- assessment of SCADA devices exposition in the entire IPv4 addresses space.

The paper is structured as follows. Section II describes related work about scanning techniques and ICS security. Section III introduces our new scanning methodology. Its efficiency is evaluated in section IV. Its application to SCADA devices and our related findings are highlighted in section V. Finally, section VI concludes our work.

II. Related work

Scanning methods are part of network-based discovery techniques. Horizontal scanning consists in testing if multiple hosts (IP addresses) expose a given service. Hence, the goal is to test if a port is open on these hosts. There are various methods which can be leveraged in that context [10], especially for TCP, and a widely used tool to perform such an activity is nmap [11]. For example, a common manner to discover TCP open ports is to try to initiate new connections with SYN-flagged

packets (TCP SYN scan). If the port is open, a SYN/ACK packet is sent back while a RST packet is representative of a closed port. ICMP replies can occur if the targeted port is filtered (ICMP port unreachable), or if the IP address is not active (Destination unreachable) but ICMP packets from external networks are usually blocked to fight scanning.

However, enumerating all IP addresses in a smart manner is a hard task. For instance, scanning IP addresses in a sequential way, *i.e.* waiting the reply back from a probed IP address or a timeout expiration before continuing cannot scale in time [12], [13]. Scanning all IP addresses simultaneously is challenging, since it supposes that the probing host maintains a state for each probed IP address until the result can be decided (response or timeout exceeded). Hence, advanced techniques have been proposed [8], [9]. Such techniques optimize the transmission of packets by using an asynchronous communication model. Whereas this speeds up the scan, low-footprint scanning requires a good randomization of targeted IP addresses to avoid testing several ones of the same subnet in a too short interval of time. Both ZMap [8] and Masscan [9] implement such a mechanism with different techniques, using respectively a multiplicative group or an encryption primitive to enumerate addresses in a random order. Actually, probing IP addresses such that consecutive ones are from different IP blocks is even better. That is why we propose in this paper a method that maximizes the distance between successively scanned IP addresses based on the longest common prefix, *i.e.* a smaller common prefix means a higher distance. From this point view, it is similar to the reverse-byte order approach proposed in [14] but with a different technique. Such an IP address enumeration technique is now also leveraged by attackers [15].

Regarding security of CPSs and ICSs, several detection mechanisms are summarized in [16]. Inspired by traditional computer security, machine learning is explored in [17]. Retrieving the behavior of a system using context-based information from traffic analysis is investigated in [18]. The authors in [19] describe a dedicated testbed for power grid allowing to test attacks and defenses. In [20], authors proposed an Intrusion Detection System (IDS) tracking the SCADA system states since these systems are complex and they are interacting with physical processes. Since all these works pave the way to secure these systems, our work is complementary to understand how these system are currently exposed in Internet. Similar to our goal, the authors in [21] use the Shodan search engine to determine the IP addresses of over 7,500 public-facing ICS devices. However, such kind of black-box service does not provide any details about the discovery methodology and may be subject to unknown bias.

III. Internet-wide scanning

A. Objectives

Performing an IPv4 scan at the Internet scale has to reach several properties:

- Property P1 - full IPv4 address space coverage: the scan has to enumerate all the 2^{32} IPv4 addresses. In practice, some IPv4 addresses are naturally excluded (local addresses, broadcast addresses,...) but the scale of the problem remains unchanged.
- Property P2 - speed: the scan has to be fast to avoid bias due to dynamic address allocation.
- Property P3 - address randomization: to avoid being blacklisted and to limit the impact on a scanned network, the rate, *i.e.* the number of tested hosts per second, on this network has to be slow. To avoid to slow down too much the speed of the scan, which is antagonist with P2, a good randomness over the IPv4 addresses is necessary, *i.e.* avoiding that consecutive scanned IP addresses belong to the same subnetwork.
- Property P4 - unpredictable address generation sequence: while the previous property guarantees that addresses are well randomized over a single scan campaign, consecutive scan campaigns needs also to be different regarding the sequence of IP addresses, to avoid the scanning tool being fingerprinted when re-used through multiple scanning campaigns.

B. ZMap

Regarding P2, ZMap [8] relies on multiple optimizations (half SYN scan, raw sockets, no state per IP address). They have been shown to be efficient and our approach also relies on them.

The address generation algorithm of ZMap allows to fulfill property P1, and to obtain a relatively good address randomization (property P3). This algorithm iterates over a multiplicative group. The first address to probe is randomly selected as a_0 and used for generating the next ones: $a_{i+1} = r \times a_i \pmod{2^{32}}$ assuming r as a primitive root of $(\mathbb{Z}/4,294,967,311)^{\times}$, 3 by default for ZMap.

To generate a different sequence of IP addresses, changing the primitive root is necessary. In total, there are about 10^9 possible generators, each generating a distinct IP addresses sequence [8].

C. Address Randomization

WiScan relies on ZMap mechanisms for fast scanning (property P2), the randomness over consecutive IP addresses in ZMap is various over the entire sequence and can be very low. For example, with $r = 3$ and $a_i = 1$, next IP addresses generated are $a_{i+1} = 3$, $a_{i+2} = 9$, $a_{i+3} = 27$... The same observation exists in IP address space regions. Therefore, there is a risk to scan several close addresses being potentially part of a block of IP addresses allocated to a single administrative authority.

We thus propose to enumerate IP addresses so that when generating the ith IP address a_i, its distance with the previous one, a_{i-1}, is maximized (*i.e.* having the smallest common prefix). Alternating the highest bit, the $32th$ bit is sufficient to have a 0-bit longest common prefix for every a_i and a_{i-1}. Once this highest bit fixed, the distance between a_i and a_{i-2} has to be maximized but the highest bit of a_i and a_{i-2} is the same since due to the alternating process. Thus, the second

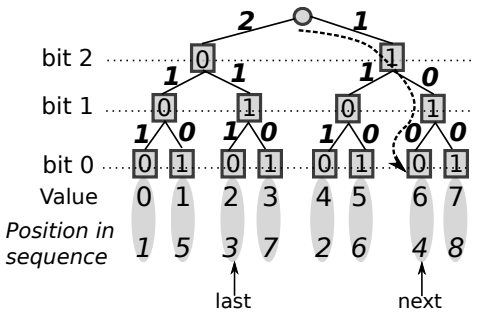


Fig. 1: Tree-based three-bits number generation.

highest ($31st$ bit) bit is reversed as well every two probed IP addresses leading to a 1-bit common prefix between them. The process continues considering a_{i-3} , a_{i-4},... with reversing the the ith bit every $32 - i + 1$ targeted IP addresses.

The generated sequence of IP addresses is equivalent to the traversal of a weighted binary tree as represented in Figure 1, where each level corresponds to one bit of the IP address, starting form the highest one, and each edge to a weight, initialized to 0 and being incremented when traversed. Each time an IP address has to be generated, the tree is traversed from the root to a leaf by following edges with the lowest weights (with the left branch as default when wights are equals). The nodes represent each bit number of IP addresses which are denoted as value in the Figure 1 but limited to 3-bits digits for sake of clarity. Assuming the last and third generated number is 2, the next is 6 following the lowest weights.

Keeping such a tree structure for all IPv4 addresses is memory consuming but this process corresponds iterating from 0 to $2^{32} - 1$ and reversing bits.

Such a methodology will generate all IP addresses (P1) with a good randomness (P3). However, the generated sequence is always the same. Considering this sequence (starting at 0) as a loop and starting with a random number (between 0 and $2^{32}-1$) would be also too weak since the order of IP addresses is always the same. This is due to a default behavior when traversing the tree and being faced with branches with the same weight.

To overcome this issue, a mask is randomly generated at the beginning. This 32-bits mask specifies the default behavior (right or left) of each tree level (*i.e.* for each new bit). In comparison with the original tree (with the left branch being selected as the default behavior in case of equality) at the same iteration of the generation of IP addresses, the weights of edge at level i will be swapped if the ith bit of the mask is 1 because all weights are initialized at 0.

As a result, the jth generated IP address differs only on masked bits with respect to the version without mask. Hence, this consists in applying a XOR operation to the generated address. Because the size of the mask is the number of bits in an IPv4 addresses, it is thus possible to have 2^{32} independent generators, while this is limited to 10^9 using ZMap. Regarding [14], the authors use a linear congruential generator such that $a_{i+1} = b \times a_i + c \pmod{2^{32}}$. With appropriate values for b and c, the randomness of generated address is similar to ours. However, our mask-based operation alleviates the need to fully regenerate the address enumeration while scanning multiple ports in parallel with different sequences. Indeed, this corresponds to applying multiple masks to a single generated sequence.

The algorithm of WiScan is summarized in 1. All addresses are generated within one loop incrementing the 32 bits value. This value is then bit-reversed and the XOR-mask is applied to get the final IP address to probe. Then the scan of this IP address is triggered. This algorithm is focused on the IP address randomization and voluntary omits practical details inherited from Zmap. In particular, it is worth to mention that the scan is asynchronous, *i.e.* so the loop is non-blocking, and non routed IP addresses like private IP addresses are discarded.

Algorithm 1 WiScan IPv4 address scanning

Require: m: a randomly-generated 32-bits mask
1: **for** $i \in 0 \dots 2^{32} - 1$ **do**
2: $addr \leftarrow reverse\_bit(i) \oplus m$
3: $i \leftarrow i + 1$
4: scan $addr$
5: **end for**

IV. Scan efficiency

To evaluate the randomness in the address sequence generated by our scanning methodology, entitled WiScan, we denote the Longest Common Prefix (LCM) between two addresses, a_i and a_j as $lcm(a_i, a_j)$. Since a higher LCM means that IP addresses are in a smaller subnetwork, and so are closer, $32 - lcm(a_i, a_j)$ is the distance between a_i and a_j.

Assuming the sequence of generated IP addresses as $S = a_0, a_1, \dots, a_{2^{32}-1}$, each IP address a_i is compared to previous ones by computing a weighted average distance:

$$d1(a_i) = \frac{\sum_{1 \leq k \leq n}(32 - lcm(a_i, a_{i-k})) \times (n - k + 1)}{n} \quad (1)$$

n controls the number of previously scanned IP address to compare with. Indeed, considering all of them is meaningless as this will result in a constant because each IP address is uniquely generated. Besides, the weight $(n - k + 1)$ gives a higher importance to distance with IP addresses generated just before the new one. This metric helps to assess how an individual address differs from those previously generated. In Figure 2, we compute $d1$ with $n = 100000$ with WiScan (our methodology) and ZMap. In the case of WiScan, the random process alternates each address bit with a regular period. Hence, the average distance $d1$ is always 18 whereas in the case of ZMap, only one half of the generated IP addresses exhibits such a distance while the other half is concentrated between 14 and 17. Therefore, our generation exhibits a better randomness since the LCM tends to be smaller when $d1$ increases. This does not represent the exact size of the subnetwork, i.e. the longest common prefix, since $d1$ is a weighted average in equation (1). However, we observed $d1$

lower than 8 for ZMap meaning that subsequent IP addresses are within small size networks with respect to the prefix size. It would be thus more visible and detectable.

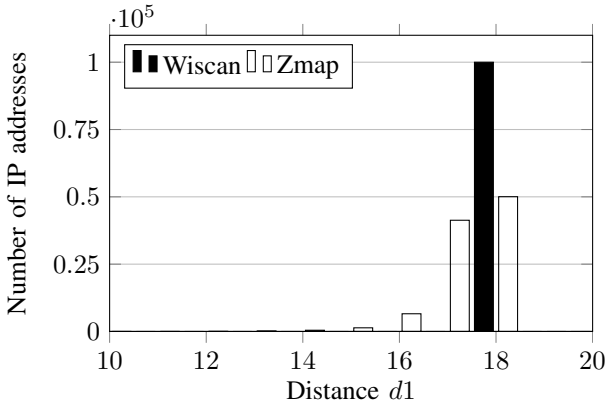


Fig. 2: Distribution of distance $d1$ over 0.25% of IPv4 addresses.

Another measure consists in evaluating how similar generated IP addresses separated by a fixed interval in the sequence (the rank) are. The rank is defined as $r(a_i, a_j) = i - j$. Assuming S as a sequence of generated IP addresses, the following distance can be thus computed assuming the rank as the main parameter

$$d2(r) = \frac{\sum_{\forall a_i \in S} 32 - lcm(a_i, a_{i-r})}{|S|} \quad (2)$$

Figure 3 introduces the resulting distance with $1 \leq r \leq 1000$. As such a distance is computed over all IP addresses, aggregated results are presented: minimum, maximum, the area between the 10th and the 90th percentile and the median. Although the median and the maximum are similar independently of the approaches (ZMap, Masscan and WiScan), $d2$ tends to be lower for Masscan for lower ranks (around 0-200). Hence, addresses scanned closely in time are thus close in the IP address space. This makes thus the scanning process more visible. Actually, avoiding low distances, even being infrequently, is of capital importance. This ability is measured by the minimum distance to be as highest as possible, making our approach the best one as shown in Figure 3.

V. Industrial Systems Exposure

In this section, we provide the details of the analysis of a first scan of the public IPv4 space using WiScan. In this scan, we targeted 5 TCP ports: 23,102, 502, 2308 and 5001. We have selected the port 23 dedicated to Telnet service to serve as a baseline and also to assess its exposition in Internet. Because it is also known to be used in many exploits, it helps in comparing the SCADA ports exposure regarding a standard port. The port 102 is used by the S7 Communication protocol for managing and programming Siemens devices and 502 is used by Schneider SCADA devices. We have also scanned other ports, 2308 and 5001, which are also known to be used by Siemens devices. Our analysis will focus on evaluating the exposure of the ports 102 and 502 in the public IPv4 address space over countries and over different network block sizes.

A. Dataset and Methodology

We performed a complete scan of the public IPv4 space. As ICS devices are sensitive, we voluntary slow down our scan for ethical reasons resulting in a scanning period of 33 days. Collected data has a size of 500MB of raw data and 17GB of enriched data. Table I summarizes the parameters of the experiment. The scan has been performed from an isolated network without any firewall restriction. Each scanning machine (eight in total) stores the results of responding hosts in a file where each entry represents a *hit* containing a positive response of an IP address and port. All result files are merged and stored in an Elasticsearch [1] suite, while enriching the IP addresses with geolocation using the MaxMind GeoIP[2] and reverse DNS resolution information. We mainly used the Kibana tool of this suite for the visualization and the extraction of several statistics from the scan results as well as Apache Pig scripting language for more advanced statistics.

Parameter	Value
Duration of the scan	33 days: from 22/07/2015 to 24/08/2015
List of ports	23, 102, 502, 2308 and 5001
Number of scanners	8 VMs
Scan rate	7 500 packets/second

TABLE I: Parameters of the scanning experiment.

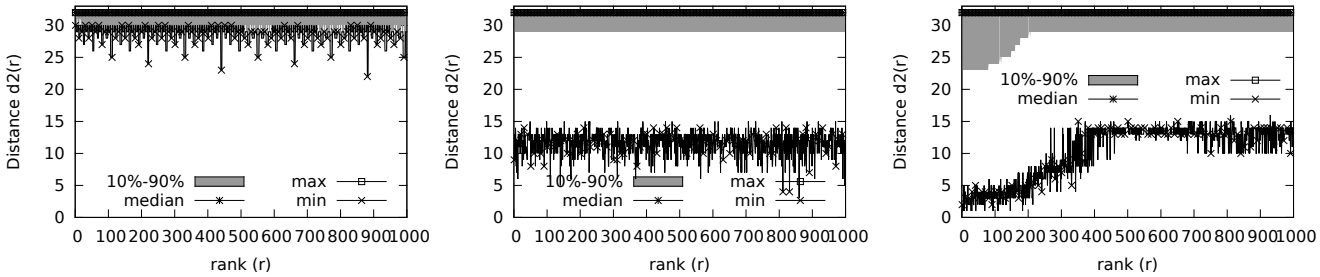
The results contain 28,823,873 hits with 19,724,400 unique IPv4 addresses (68% of all hits) since the same IP address may have multiple open ports. Table II presents the percentage of hits for each port. We mainly observe that the port 23 dedicated to Telnet service is widely available and exposed with a hit percentage of 63.75%. The Shodan Search Engine[3] also made the same observation during the year 2015 indicating that the number of hits for this port has increased. We have also found a number of hits for the port 502 similar to the Censys search Engine[4] [22]. Therefore, our scanning methodology is concordant with state-of-the art approaches.

Port	Number of hits	Hits percentage
23	18,374,318	63.74%
102	2,441,676	8.47 %
502	2,461,881	8.54%
2308	2,430,182	8.43%
5001	3,115,808	10.80%

TABLE II: Number of hits by port.

We firstly analysed the distribution of the number of open ports by IP address. The results are depicted in Figure 4 where each bar is representative to a set a IP addresses among which between 1 and 5 ports are reported as open. We have considered 3 datasets in this analysis. The first dataset contains

[1] http://www.elastic.com
[2] https://www.maxmind.com/
[3] https://www.shodan.io/
[4] https://www.censys.io/ipv4?q=modbus.502



(a) WiScan (b) ZMap (c) Masscan

Fig. 3: Distance between generated IP addresses.

all the scanned ports. In the second dataset, we excluded the port 23 and in the third dataset we only considered the two ports 102 and 502.

Assuming the first dataset, 86% of IP addresses have only a single accessible port and the percentage with a number of ports between 2 and 5 is very low: 10% for 5 ports on the same IP address and under 1% for 2, 3 and 4 ports.

For the second dataset, when excluding the port 23, the total number of IP addresses decreases to 3,457,232 leading so to 29% and 64% of them having respectively one and four ports open on the same IP address. Hence, when two ports are discovered as open, there is a higher chance to have also other ports open.

For the third dataset with the two specific SCADA ports 102 and 502, 89% of their total responding IP addresses (2,581,213) have these two ports co-jointly open. However, only 10% (260,344) of the total IP addresses in this set have a single open port either 102 or 502.

Because these ports are representative of field devices from different manufacturers (Siemens and Schneider), having them open on the same field device is impossible. After having investigated some of these IP addresses, this result is mainly due to different devices accessible through a common gateway, which exposes multiple ports. However, only an intrusive scan where application-layer data is retrieved by performing a full connection could confirm it. For ethical reason, we refrain ourself from achieving this. Indeed, such full connections attempts might be considered as attacks and above all could really disturb these field devices, since many of them have weak TCP/IP stack implementations.

Therefore, we only consider the 10% of devices with a single open port either 102 or 502 as directly exposed field devices.

In the next sections, our analysis is focused on ports 102 and 502 which are common ports for Siemens and Schneider devices. The ports 5001 and 2308 will not be integrated in our analysis to limit false positives, since they are also used by other IT services and not only by SCADA systems.

B. Geographic exposure

Using the enriched dataset with GeoIP informations, we found that the ports 102 and 502 are respectively exposed

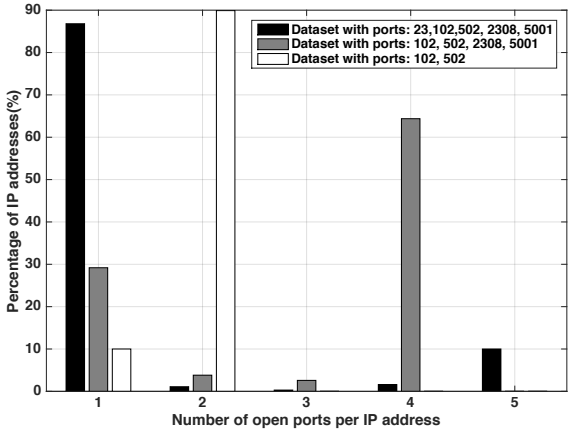


Fig. 4: Percentage of IP addresses per number of open ports regarding the total number of IP addresses having at least one open port.

in 230 and 232 countries. This result is not surprising since Siemens and Schneider devices are widely adopted and represent top sellers of SCADA solutions.

Figure 5 shows the top 25 countries per the number of hits for the two ports. We mainly observe that the United States has a large number of hits, around 1,500,000. China and Belgium have respectively also large number of hits, around 600,000. We also observe that the number of hits is equally divided between the two ports, since as we stated above, 89% of the hits share the two ports (Figure 4).

Then, we were interested in the Empirical Cumulative Distribution Function (ECDF) of the number of hits by port in the scanned countries in order to assess its geographic discrepancy. The results are depicted in Figure 6. We mainly observe that the two ports 102 and 502 have closer ECDF. For these ports, 20% of the countries have 1 hit on each on them and 90% of the countries have a number of hits under 10,000. Thus at the global scale, the number of hits for SCADA ports is relatively small. As shown, 10% of the most exposed countries are exposed with a level hundred times higher than others (from 10^4 to 10^6).

This coherent with our previous observation stating that only

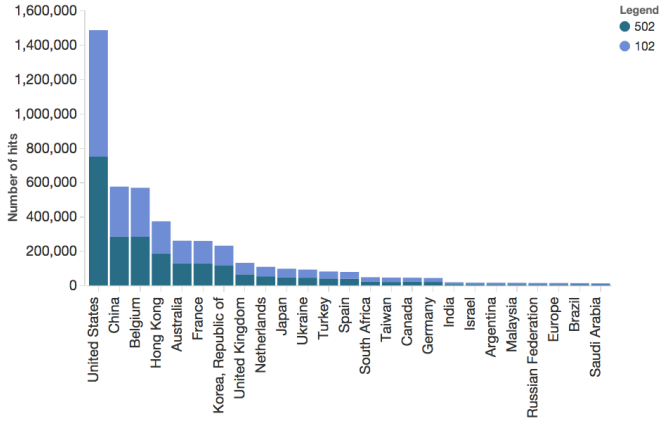


Fig. 5: Top 25 countries of open SCADA ports (102 and 502).

few countries (3%), mainly United States, China, Belgium, South Korea, Hong Kong, France and Australia are responsible for the majority of them. This could be explained by a conjunction of factors: larger proportions of industrial installations in these countries, especially old ones with less security, and good Internet connectivity.

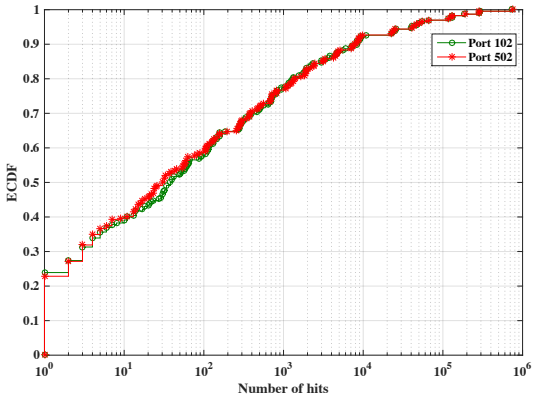


Fig. 6: ECDF of the number hits in the scanned countries.

C. Subnetworks exposure

In this part, IP addresses are aggregated per subnetworks assuming /8, /16 and /24 prefix sizes. We first identified the number of subnetworks having at least one IP address with an exposed port to be compared with the total number of existing subnetworks. The results are presented in Table III and highlight that around 76% of the /8 subnetworks have at least one hit. The two major SCADA ports (102 and 502) have an exposition percentage around 22%-23%, in terms of the number of networks, in the /16 network space. This percentage is around 0.5% in the /24 network space .

We also investigated the distribution of the number of IP addresses with an open port in the /8, /16 and /24 network blocks as depicted in Table IV (considering only those having at least one IP address with the port open). In the /24 network block, the number of subnetworks exposing a single port is

	Number of subnetworks		
Network block size	/24	/16	/8
Port 102	94,559	15,229	195
Port 502	96,208	14,564	196

TABLE III: Number of subnetworks having at least one IP address exposing a port.

lower than the number of subnetworks exposing two open ports. In /16 space, the number of subnetworks exposing the two ports is mostly twice the number of subnetworks exposing a single port. Finally, in /8 space only 1 subnetwork is exposing a single port and 195 subnetworks are exposing both of them.

	Number of subnetworks		
Network block size	/24	/16	/8
1 port	50,817	6,443	1
2 ports	69,975	11,675	195

TABLE IV: Number of subnetworks by the number of exposed ports.

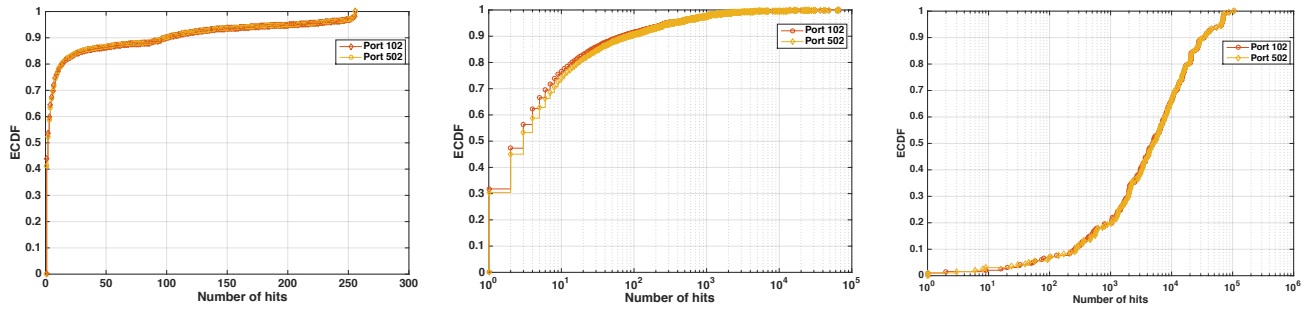
Then, we investigate how many hits we have in each of these subnetworks by network block as depicted in Figure 7. For /24 networks as shown in Figure 7(a), we observe that 42% of the networks have a single hit for the two ports 102 and 502. We observe also that 85% of the responding networks have up to 90 hits, meaning that 15% have at least 90 hits. This is a high value in comparison with 255, the total number of addresses in an individual /24 network. Regarding all the scanned ports, we found that around 1660 /24 networks have all five ports open on all IP addresses and can be supposed to be considered as honeypots or misconfigured.

In /16 networks, as shown in Figure 7(b), the fraction of networks with a single hit is around 30% and it is close to 88% for 100 hits. In /8 networks (Figure 7(c)), the fraction for low number of hits is very low due to the aggregation. For SCADA specific ports (102 and 502), 20% of the networks have less than 1000 IP addresses concerned, and this value is always below 100,000.

Therefore, ICS devices are concentrated in rather small networks (/24) but the latter are well scattered over Internet, i.e. in the IPv4 space. From a security point of view, /24 networks could be more easily disrupted by attacks like Denial-of-Service than large distributed networks. Hence, this kind of attack against /24 network would have a high impact by affecting multiple ICS devices meantime.

VI. Conclusion and Future Work

In this paper, we introduced WiScan, our methodology for Internet-Wide scanning performed with a modified version of the ZMap tool to avoid targeted networks to be overloaded. The main modification concerns the random generation of IPv4 addresses. We compared our technique in terms of the distance between successive scanned IP addresses with the original version of ZMap and the Masscan tool. A first scan has been performed by targeting 5 distinct ports (23, 102, 502,



(a) /24 network blocks (b) /16 network blocks (c) /8 network blocks

Fig. 7: ECDF of number of hits for /8, /16 and /24 network blocks.

2308 and 5001). During this scan, only two complaints have been addressed to us by administrators of targeted networks. From our scanning campaign, we can claim that the exposure of SCADA systems is a reality, especially in industrialized countries (United States, China, Belgium, France, South Korea), and it is possible to identify specific subnetworks with many accessible addresses on SCADA specific ports.

In future work, we will rely on the results of multiple conducted scans to develop a novel methodology to fingerprint scanning activities, mainly to be able to detect low profile and persistent scans usually achieved by attackers to target vulnerable hosts.

ACKNOWLEDGEMENT

This work was partially funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its FP7 Programme, and by HuMa, a project funded by Bpifrance and Region Lorraine under the FUI 19 framework. It is also supported by the High Security Lab hosted at Inria Nancy Grand Est (http://www.lhs.loria.fr).

CODE RELEASE

The source-code of WiScan is expected to be released with open-source license soon on the following project web-page: http://wiscan.gforge.inria.fr. It is currently under an ethical review due to the potential malicious use of WiScan.

REFERENCES

[1] J. Weiss, *Protecting industrial control systems from electronic threats*. Momentum Press, 2010.

[2] S. McLaughlin, "Securing Control Systems from the Inside: A Case for Mediating Physical Behaviors," *IEEE Security & Privacy*, vol. 11, no. 4, pp. 82–84, 2013.

[3] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security & Privacy*, vol. 9, no. 3, pp. 49–51, May 2011.

[4] B. Miller and D. Rowe, "A survey of SCADA and Critical Infrastructure Incidents," in *Proceedings of the 1st Annual conference on Research in information technology*. ACM, 2012, pp. 51–56.

[5] E. J. M. abd Hary A. Waxman, "Electric Grid Vulnerability: Industry Responses Reveal Security Gaps," http://democrats.energycommerce.house.gov/sites/default/files/documents/Report-Electric-Grid-Vulnerability-2013-5-21.pdf, May 2013.

[6] I. Fovino, A. Carcano, M. Masera, and A. Trombetta, "Design and implementation of a secure modbus protocol," in *Critical Infrastructure Protection III*, ser. IFIP Advances in Information and Communication Technology. Springer, 2009.

[7] K. A. Stouffer, J. A. Falco, and K. A. Scarfone, "Sp 800-82. guide to industrial control systems (ics) security: Supervisory control and data acquisition (scada) systems, distributed control systems (dcs), and other control system configurations such as programmable logic controllers (plc)," Gaithersburg, MD, United States, Tech. Rep., 2011.

[8] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *USENIX Security Symposium*, Washington, D.C., 2013.

[9] R. Graham, "MASSCAN: Mass IP port scanner, accessed on 07/01/2016," https://github.com/robertdavidgraham/masscan.

[10] M. de Vivo, E. Carrasco, G. Isern, and G. O. de Vivo, "A review of port scanning techniques," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 2, pp. 41–48, Apr. 1999.

[11] G. F. Lyon, *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure, 2009.

[12] N. Provos, P. Honeyman, and P. Honeyman, "ScanSSH - Scanning the Internet for SSH Servers," in *Proceedings of The 15th USENIX Systems Administration Conference*, 2001, pp. 25–30.

[13] P. Eckersley and J. B. A. observatory for the SSLiverse, "An Observatory for the SSLiverse," https://www.eff.org/files/DefconSSLiverse.pdf, 2010.

[14] D. Leonard and D. Loguinov, "Demystifying internet-wide service discovery," *Transactions on Networking*, vol. 21, no. 6, pp. 1760–1773, Dec. 2013.

[15] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapé, "Analysis of a "/0" stealth scan from a botnet," *Transactions on Networking*, vol. 23, no. 2, pp. 341–354, Apr. 2015.

[16] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *Comput. Surv.*, vol. 46, no. 4, pp. 55:1–55:29, Mar. 2014.

[17] F. Schuster, A. Paul, and H. König, "Towards learning normality for anomaly detection in industrial control networks," in *Conference on Autonomous Infrastructure, Management, and Security (AIMS)*. Springer, 2013.

[18] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer, "Semantic security analysis of scada networks to detect malicious control commands in power grids," in *Smart Energy Grid Security (SEGS)*, ser. SEGS. ACM, 2013.

[19] A. Hahn, B. Kregel, M. Govindarasu, J. Fitzpatrick, R. Adnan, S. Sridhar, and M. Higdon, "Development of the powercyber scada security testbed," in *Cyber Security and Information Intelligence Research (CSIIRW)*. ACM, 2010.

[20] A. Carcano, I. Fovino, M. Masera, and A. Trombetta, "State-based network intrusion detection systems for scada protocols: A proof of concept," in *Critical Information Infrastructures Security*, ser. Lecture Notes in Computer Science. Springer, 2010, vol. 6027.

[21] E. P. Leverett, "Quantitatively assessing and visualising industrial system attack surfaces," *University of Cambridge, Darwin College*, 2011.

[22] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, "A search engine backed by internet-wide scanning," in *SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2015.