CSEP564 : Computer Security : Reading 5

Karuna Sagar Krishna

October 30, 2024

Paper Title

Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices

Paper Authors

Nadia Heninger, Zakir Durumeric, Eric Wustrow, J. Alex Halderman

Problem

The author aims to provide a comprehensive Internet scale evaluation of the impact of using weak random number generators in two cryptographic protocols - TLS and SSH. These protocols use RSA and DSA primitives to secure the communication channel and it is well known that these primitives fail when used with a malfunctioning random number generators (RNG). Problems with faulty randomness are classified and further investigated to identify subtle problems which are more evident at Internet scale.

Approach

The authors perform the comprehensive evaluation by scanning the entire Internet (IPv4 address space) looking for standard TLS and SSH open ports. For each TLS host, the author performed TLS handshake to capture the certificate presented by the service/host. This certificate was then parsed to extract useful information and stored in a database. For each RSA based SSH hosts, the host keys were collected and stored. For each DSA based SSH hosts, two scans were done with varied client string to collect authentication signatures in both cases to identify repeating ephemeral keys. And finally to identify vulnerable devices, the authors employed a combination of TCP/IP finger printing, examining certificate fields and the hosts website contents.

With the data collected, the authors performed various analysis to identify and collect statistics about vulnerabilities. These vulnerabilities can be classified into 3 buckets - repeated keys, factorable RSA keys and DSA signature weakness. Repeated keys could be legit when used by in shared hosting situation or when they all belong to the same organization. Vulnerable repeated keys can be either due to use of default keys shipped by manufacturer or that keys were generating using low entropy RNG. The authors developed a quasi-linear algorithm to compute all-pairs GCD identifying RSA keys that same common factors; interestingly this Internet scale computation costed $5! DSA signature weakness was identified by looking at the value of r in two authentication signature captured earlier.

The authors performed experiments and source code analysis on 3 popular open source implementations used as building blocks in TLS and SSH services - Linux kernel, OpenSSL and Dropbear. The paper describes various entropy pools maintained by Linux kernel, how it sources entropy and shows data on how entropy is accumulated during boot time. In

short, usability issues could lead to seeding with low entropy at boot time. OpenSSL use of RNG and entropy pools explains why factorable RSA key vulnerabilities exist in the wild. The authors prove their hypothesis by dilating time and showing that a slower hardware or faulty clock can lead to factorable RSA keys. Finally, Dropbear SSH server had faulty implementation in maintaining its entropy pool that could be exploited to trick multiple hosts to generate same ephemeral and long term private keys.

Conclusions

The authors conclude that weak keys are a result of specific design and implementation choices rather than cryptographic weakness. The Internet scale scan offered a macroscopic view to identify vulnerabilities that were not possible to be identified otherwise. Hence the authors hope this approach can be applied more broadly to mine for vulnerabilities hiding in plain sight.

New Ideas

The authors derive a quasi-linear algorithm to compute all-pairs GCD. This is great idea that backs their online key checking service allowing users to test if their keys are vulnerable. Interestingly this algorithm is fast and cheap and the service takes a step in making it accessible to users.

The paper provides targeted and practical suggestions to various stakeholders in the ecosystem. These actionable suggestions condensed from various experiments and analysis help improve security posture across the stack.

Improvements

The paper does a great job of identifying and quantifying vulnerabilities. However, it does not clarify the real world risk of these vulnerabilities. Adding a section to highlight the concrete risk, listing or referencing real world attacks would help highlight the seriousness.

The authors could have open sourced their code. This allows open source community to enhance and maintain the tools which can be applied more broadly. This also avoid (for example) - currently the online key checking service is no longer available.

New Directions

The number of ubiquitous computing devices have grown dramatically in the last decade - automobiles, phones, smart cards, utilities. We need to study how RNGs work in these devices and identify their vulnerabilities and impact.

Various cloud companies have devices and services on closed network not reachable from Internet. Yet, these devices and services are susceptible to vulnerabilities listed in this paper. We should employ techniques developed in this paper and build tools that can continuously scan these services. Use of AI/ML agents to automatically detect patterns at large scale could be beneficial and needs more research.