ICS vulnerability - ICS Advisory (ICSA-22-104-05) Siemens OpenSSL Vulnerabilities in Industrial Products

Om Tatipamula
George Mason University
Fairfax, Virginia, United States

Aidan Shaugnessy
George Mason University
Fairfax, virginia, United States

Mahish Mahendarkar
George Mason University
Fairfax, virginia, United States

Innocent Green
George Mason University
Fairfax, virginia, United States

ABSTRACT

The object of this paper is to explore a NULL pointer dereference vulnerability that has been identified in SIEMENS ICS products using the OpenSSL service.The paper explores four papers covering topics related to all aspects of this vulnerability(identification, remediation, best practices). The first paper discusses basic recommendations for owners and operators of critical infrastructure. The second paper discusses the Null Pointer Dereference vulnerability. The third source discusses a collection of commonly found vulnerabilities in the software design process. The fourth paper discusses the NULL pointer vulnerability in detail with code excerpts and specific mitigation strategies compliant with current international vulnerability mitigation standards.

ACM Reference Format:
Om Tatipamula, Aidan Shaugnessy, Mahish Mahendarkar, and Innocent Green. 2018. ICS vulnerability - ICS Advisory (ICSA-22-104-05) Siemens OpenSSL Vulnerabilities in Industrial Products. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Siemens has been around for nearly 175 years as a global innovator. They have focused on digitalization, and are a leader in power generation and distribution, intelligent infrastructure, and distributed energy systems. The company has developed technologies that support multiple American industries including manufacturing, energy, healthcare, and infrastructure. With industrial control systems in Siemens industrial products being mostly digitized now, along with that comes many vulnerabilities and risks. Just like the exploit that was posted publicly on April 14th.

The Executive summary of the attack is the following; with the vulnerability title of CVSS v3.5.9. To bring attention to the following vulnerability CISA labels the attack as high attack complexity but it can be exploited remotely. This attack has only affect Siemens industrial products and the vulnerability is the “NULL Pointer Dereference”. Successful exploitation of this vulnerability may allow an unauthenticated attacker to cause a denial-of-service condition if a maliciously crafted renegotiation message is sent. How exactly does the “NULL POINTER DEREFERNCE CWE-476” work? Well an OpenSSL TLS server may crash if sent a maliciously crafted renegotiation ClientHello message from a client. If a TLSv1.2 renegotiation ClientHello omits the signature algorithms extension, where it was present in the initial ClientHello, but includes a signature algorithms cert extension, then a NULL pointer dereference will occur, leading to a crash and a denial-of-service condition. A server is only vulnerable if it has TLSv1.2 and renegotiation enabled, which is the default configuration. OpenSSL TLS clients are not impacted by this issue. All OpenSSL 1.1.1 versions are affected by this issue. Users of these versions should upgrade to OpenSSL 1.1.1k. OpenSSL 1.0.2 is not impacted by this issue. This vulnerability is fixed in OpenSSL 1.1.1k. Although the following vulnerability has been fixed in one of the latest hotfixes, this vulnerability opens the door for malicious users to cause possibly even more denial of service in the future.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1.1 Problem Statement

With the thought of the vulnerability that has been discussed being a gateway for malicious users to cause possibly even more denial of service in the future, we must conduct a plan that could possibly minimize the risk of the attacks. The point of this paper is to identify and to take defensive measures to minimize the risk of exploitation of this vulnerability, by minimizing network exposure for all control system devices. We can do that through many mitigation strategies and risk assessments as defensive measures. The idea is to minimize network exposure for all control system devices and/or systems, and ensure they are not accessible from the Internet. Also locate control system networks and remote devices behind firewalls and isolate them from the business network. As well as when remote access is required, use secure methods, such as Virtual Private Networks (VPNs), recognizing VPNs may have vulnerabilities and should be updated to the most current version available. Also recognize VPN is only as secure as its connected devices.

2 COMPARATIVE ANALYSIS/RELATED WORK

In this section we will be discussion the similarity and differences of the research sources that we have found. This paper will also discuss the related work that is related to the vulnerability of the NULL pointer deference.

2.1 CISA

CISA is a government source that has gone over targeted cyber intrusion detection and mitigation strategies. This paper was developed by the ICS-CERT to enhance their network security posture. The Industrial Control Systems Cyber Emergency Response Team (ICS-CERT) provides a control system security focus in collaboration with US-CERT to conduct vulnerability and malware analysis, provide onsite support for incident response and forensic analysis, provide situational awareness in the form of actionable intelligence, coordinate the responsible disclosure of vulnerabilities/mitigations, and share and coordinate vulnerability information and threat analysis through information products and alerts.

The paper highlights that targeted cyber instructions have increased in recent months against owners and operators of industrial control systems across multiple critical infrastructure sectors. This guidance and mitigations that ICS-CERT provides are those for networks who have been compromised by cyber intrusion attacks and organizations whose desire to improve their network security. CISA even provides guidance and recommended practices, although these practices are non specific to the vulnerability that our paper is talking about it still provides powerful insight on how to prevent cyber intrusions that are like the OPENSSL vulnerability. CISA states that The impacts of a cyber intrusion will likely be different for every organization depending on the nature of the compromise and the organization's capabilities to respond. Each organization must assess its particular situation, identify the criticality of the impacted devices, and develop a prioritized course of action. Unfortunately, a simple and prescriptive remedy that can be applied uniformly to every organization does not exist. However, basic principles and recommendations exist that are essential to maintaining a sound network security posture and that will provide the necessary capabilities to respond to an incident.

CISA, with the guidance and recommended practices, also dives deep into what mitigations can be made for industrial control system industries.

- Proper Permission Management: Establish an appropriate privileged account hierarchy for administrative accounts. In a proper hierarchical design administrative rights and administrative responsibility are inversely proportional to each other. For example, Domain Administrators should only be used to administer the domain controllers, while a help desk account should have few administrative rights. This design approach should also include decisions about which hosts will allow the accounts and the manner in which the administrator accesses the devices.
- Network/System Design and Policies
  - Principle of Internet: Use infrastructure devices and software to create security zones that group users who need to communicate with each other. This helps to slow or prevent an intruder's lateral network movement. Use host-based firewalls to restrict incoming connections as another method for impeding unneeded inter-host communication.
  - Exercise Caution: IT administrators should consider disabling or removing local machine accounts, or at least ensure that local accounts across the network have unique passwords this is because on all devices could share a baseline therefore they would share the same password
  - Reboot: All machines be rebooted immediately after being used by a privileged user. The reboot process clears the user's credentials from memory, a common target of pass-the-hash tools.
  - Multi-Factor authentication: Organizations should consider moving to a multi-factor authentication system or at least ensure users choose complex passwords that change regularly.

2.2 MITRE

The source from MITRE is a listing on the Common Weaknesses Enumeration (CWE). CWE is a "community-developed list of software and hardware weakness types." This provides a "common language" to discuss common weaknesses found in software and hardware, a "measuring stick for security tools" which can be used to quantitatively analyze weaknesses against each other, and a "baseline for weakness identification, mitigation, and prevention" which can be used by organizations across the globe to tighten their security architecture.

Currently over 900 weaknesses have been identified by CWE and the list is expanding every year. The large number of weaknesses shows that it is more like an encyclopedia than a tool which security experts should use in their day-to-day work. For example, the Seven Pernicious Kingdoms has a great real world use in helping software developers write secure code. Although the entire encyclopedia cannot be practically applied to an organization we can use specific entries to learn about and mitigate specific weaknesses.

For the purposes of the Siemens vulnerability we are interested in CWE-476: "NULL Pointer Dereference". According to the description, this CWE occurs when "the application dereferences a pointer that it expects to be valid, but is NULL, typically causing a crash or exit." The cause of the dereference can be attributed to "a number of flaws, including race conditions, and simple programming omissions."

The CWE classification system has heirachical characteristics where some entries are children/parents of other entries. CWE-476 in particular is the child of CWE-754 and CWE-710. The former is the "Improper Check for Unusual or Exceptional Conditions" and can occur when "[t]he programmermay assume that certain events or conditions will never occur or do not need to be worried about". The latter is the "Improper Adherence to Coding Standards" which occurs when "[t]he software does not follow certain rules for development". CWE-476 is the parent of CWE-690 which is when "[t]he product does not check for an error after calling a function that can return with a NULL pointer if the function fails". CWE-690 can result in a NULL pointer dereference which is why it is the child of CWE-476.

There are two scopes for the impact of a NULL pointer dereference. The first is when the scope is limited to just availability and this can be seen through a Denial of Service caused by a crash, exit, or restart. This is the most common outcome of NULL Pointer dereference because exception handling may not be implemented. And even in cases where exception handling is being used it may be difficult to return the application to a safe state. The second scope can impact all components of the CIA triad: Confidentiality, Integrity, and Availability. The technical impact of such a NULL pointer exception is the execution of unauthorized code, the ability to read unauthorized regions of memory, and the ability to write to unauthorized regions of memory. This is a very rare consequence of NULL pointer dereference and can only occur when "NULL is equivalent to the 0x0 memory address and privileged code can access it, then writing or reading memory is possible, which may lead to code execution."

The vulnerability which affected Siemens was only able to result in a Denial of Service and not code execution.

2.3 ImmuniWeb

This advisory bulletin discusses the CWE-476 which has been designated as the NULL pointer dereferencing weakness in the CWE list where CWE is defined as a community-developed list of software and hardware weakness types. The NULL pointer dereferencing error affects primarily C/C++/Assembly as these languages make use of pointers. When the value of a memory location is obtained by a pointer the pointer is classified as dereferenced.

The error occurs when this derefernce occurs for an invalid address and a NULL location. This is a logical error that can be exploited via race condition. This error typically crashes the entire application, but the potential for a DOS attack or privilege escalation via foreign code execution opens up. The risk of this occurring may not seem to be very high but the criticality of such errors is crucial when considering ICS networks in control of utilities which affect the day to day lives of a multitude of individuals.

An example of vulnerable code is presented below.

```
// NULL Pointer Dereference [CWE-476] vulnerable code
example // (c) HTB Research
undef UNICODE include "StdAfx.h" include <winsock2.h>
include <ws2tcpip.h> include <stdio.h>
int cdecl main(int argc, char **argv)  WSADATA wsaData;
int iResult; INT iRetval;
DWORD dwRetval;
int i = 1;
struct addrinfo *result = NULL;
struct addrinfo *ptr = NULL;
struct addrinfo hints;
if(argc<2)
printf("usage:
return 1;

iResult = WSAStartup(MAKEWORD(2, 2), wsaData);
if (iResult != 0)
printf("WSAStartup failed:
return 1;
ZeroMemory( hints, sizeof(hints) );
hints.ai_family = AF_UNSPEC; hints.ai_socktype = SOCK_STREAM;
hints.ai_protocol = IPPROTO_TCP; dwRetval = getaddrinfo(argv[2], argv

if ( dwRetval != 0 )
printf("getaddrinfo failed with error:
WSACleanup();
```

```
return 1;
printf("getaddrinfo returned success");
return 0;
```

This error can be mitigated via the following solutions: sanity checks on pointers that are used, checking results of a return value to verify it is not NULL and consequently applying error handling, input sanitization on variables and data stores, explicit initialization of variables during declaration/before first usage, ensure proper 'locking' APIs lock conditional statements especially when multi-threads are in use. The following is an example of the logic involved with proper pointer validation [if(pointer1 != NULL) then free(pointer1) and set pointer1 to NULL].

2.4 Seven Pernicious Kingdoms

This is a paper authored by Kartina Tsipenyuk, Brian Chess, and Gary McGraw. The paper is titled "Seven Pernivious Kingdoms: A Taxonomy of Software Security Errors". The goal of the paper was to "help developers and security practitioners understand common types of coding errors that lead to vulnerabilities." They organized these errors into a "simple taxonomy" of 7+1 groups to make it easier to understand. Each Kingdom has subcategories known as Phylum which are defined as "a specific type of coding error". Phylum sharing a common theme are put into a Kingdom. For example, "Null Dereference" is a phylum from the Code Quality Kingdom.

The authors believed that software developers "play a curcial role in building secure computer systems." This is because "roughly half of all security defects are introduced at the source code level". Thus this paper is a great resource in securing our cyber landscape by identifying and fixing issues at the lowest level possible.

Most of the previous research done in existing classification schemes have been "focusing on making the scheme deterministic and precise". This means they try to achieve a "one-to one mapping between a vulnerability and the category the vulnerability belongs to". This level of precision is largely unnecessary for the purposes of helping developers write secure source code. And in some cases it may even detract from that goal. Another difference in previous research is that "[m]ost of the proposed schemes focus on classifying operating systems-related security defects rather than the errors in software security." This is a big issue because software security is an overarching concept that can be applied regardless of the operating system being used; it is independent of the platform.

To be reach the widest audience of software developers and to show them how to develop secure code the team behind this paper had to take a different approach than the ones of previous researchers. They chose "practical language centered on programming concepts" rather than "the breadth and complexity essential to theoretical completeness." This is an important step in the evolution of classification systems since the theoretical completeness of a classification scheme is not relevant if its practical applications are few and far between.

The classification scheme provided by the authors is one that is future forward. It is not all-encompassing but is intended to be "open-ended and amenable to future expansion." They "expect the list of important phyla to change over time". The important kingdoms are also expected to change over time but hopefully at a lesser rate.

Although we may be security experts, the goal of this taxonomy is to be intuitive to "software developers who are not security experts". This is why we had to read and analyze this paper through a different paradigm.

Another important feature of this classification scheme is that all of the phyla are detectable through static code analyzers such as PVS-Studio and CPPChecker. This means that software developers are able to keep track of the security of the code they write without consulting with the security team at every step of the process. Note this is great for quick feedback on the code but in reality the security team should be involved in the development of source code.

We are only interested in the Null Dereference portion of this taxonomy but it was so educational that the entire team decided to familiarize ourselves with ALL kingdoms and phyla. The seven kingdoms are:

- "Input Validation and Representation" which includes "[s]ecurity problems result[ing] from trusting input"
- "API Abuse" is when the contract between a caller and a calleee is violated
- "Security Features" is "concerned with topics like authentication, access control, confidentiality, cryptography, and privilege escalation"
- "Time and State" is about "defects [related] to unexpected interactions between threads, processes, time, and information."
- "Errors" has to do with "handling errors poorly" or "producing errors that either give out too much information [or] are difficult to manager"
- "Code Quality" has to do with "[p]oor code quality [that] leads to unpredictable behavior."
- "Encapsulation is about drawing strong boundaries"

There is an eighth, unnumbered kingdom called "Environment" which contains "everything that is outside of the source code but is still crucial to the security of the product that is being created".

The portion of this scheme that we are concerned with for the purposes of the Siemens vulnerability is the "Null Dereference" phylum from the "Code Quality" Kingdom. It occurs when "[t]he program can potentially dereference a null pointer, thereby raising a null `NullPointerException`."

3 RESULTS & CONCLUSION

In conclusion, the NULL derefercing error has been patched by SIEMENS and a NULL pointer dereferencing mitigation strategy has been discussed. The development process of software should involve proper testing and secure software designing while understanding the limitations and potential issues associated with the programming language and infrastructure that ICS systems rely on. In addition to conducting code reviews, ICS systems should be secure according to the defensive measures outlined by CISA: Hierarchical privileges, Risk assessment prior to admin control over own systems, Restrict SeDebugPrivilege (DLL vuln), Network/System design, Internet/DMZ/intranet, Baseline image potential risk (shared password local machines), Reboot after privileged user usage, No LAN Manager hash(legacy consideration),MFA(smart cards/regular password updates). Simple actions such as network segmentation and vigilant log keeping can ensure that if a 3rd party service such as the OpenSSL certificator do have logic errors the entire ICS system has not been opened to potential attackers.

REFERENCES

[1] CISA Cisa.gov. 2022. Targeted Cyber Intrusion Detection and Mitigation Strategies (Update B) | CISA. [online] Available at: <https://www.cisa.gov/uscert/ics/tips/ICS-TIP-12-146-01B> [Accessed 7 May 2022].

[2] MITRE Cwe.mitre.org. 2022. CWE - CWE-476: NULL Pointer Dereference (4.7). [online] Available at: <https://cwe.mitre.org/data/definitions/476.html> [Accessed 7 May 2022].

[3] ImmuniWeb Immuniweb.com. 2022. NULL Pointer Dereference Vulnerability | CWE-476 Weakness | Exploitation and Remediation. [online] Available at: <https://www.immuniweb.com/vulnerability/null-pointer-dereference.htmlseverity> [Accessed 7 May 2022].

[4] Seven Kingdoms Tsipenyuk, K., Chess, B. and McGraw, G., 2005. Seven Pernicious Kingdoms: A Taxonomy of Software Security Errors. IEEE Security and Privacy Magazine, 3(6), pp.81-84.