Assignment 1

CS 458 Computer Security and Privacy

Question 1: Comparison of Traditional Voting and Internet Voting (18 marks)

Voting Flows

Traditional:

1. Voter registration (where officials ensure that a voter is eligible to vote and has not voted yet)
2. Voter receives paper ballot.
3. Voter fills in and submits ballot.
4. Votes are kept in locked ballot boxes.
5. Officials count votes and phones headquarters.
6. Signed report is sent to headquarters.

Internet:

1. Voter receives voting link and authorization code.
2. Voter uses electronic devices to vote via internet.
3. Votes are kept electronically in centralized server.
4. Server aggregates the votes and communicates result.

Assumptions

In general, paper-based voting exhibits few vulnerabilities due to its time-tested nature, and involvement of physical elements and human agents. This conclusion, however, is drawn based on several assumptions:

1. It is easier to fool a program than it is to fool a human.
2. Breaking a physical lock is punished more severely than breaking a digital lock.
3. There is no systematic corruption in voting officials.

We also make the following assumptions about digital systems:

1. Well implemented end-to-end encryption cannot be cracked in a reasonable length of time.
2. It may be possible to build formally verifiable algorithms and systems theoretically, they rely on many assumptions about physical security, hardware guarantees, etc. Thus, they cannot be perfect in the real world.

Threats to Paper-based voting

1. Fabrication

Fabrication is perhaps the biggest threat to a paper-based voting system. Voter registration can be exploited by employing look-alikes, forging identity documents, and fabricating voting

letters. Once the voter is successfully registered, there are no further security mechanism to prevent a fraudulent vote. This, however, only occurs on individual basis, and is difficult to replicate with scale. A more impactful scenario can be caused by insider manipulation of vote counts, but it will require significant corruption within the voting officials.

2. Interception

In a properly run balloting center, interception of votes will be difficult. The voting officials who have control over the locked ballot boxes, however, have full access to the ballots. This implies that clever "human engineering" could lead to leak of sensitive information (such as who voted for which at what time), threaten general privacy of the voters, and enable voter coercion.

3. Interruption

Interruption of the voting procedure or data could be desirable for attackers to cripple the democratic process. Physical theft of the voting letters distributed to the household could severely reduce voter turn up, and thus, skew the final poll. Voting officials, if corrupted by the attackers, could under-report the actual voter turnout and/or the vote counts.

4. Modification

In a similar line of reasoning, corrupted officials can also be controlled to modify the vote counts entirely. Furthermore, a man in the middle attack might be possible. An attacker can pretend to be the headquarters to the voting officials to intercept phone call and signed report from voting districts. Subsequently, he or she can phone the headquarters to communicate a modified vote count, and later modify and send the signed report.

Threats to Internet voting

1. Fabrication

It may be difficult to fabricate fake identities that are verifiable through an electronic system, but attackers can target the very systems that administers voting link and authorization codes. For example, attackers can modify the system to hand out codes for recently moved or deceased individuals. Thus, fictitious voters can be created to participate in the voting process and skew the election result.

2. Interception

Although multi-party computation and homomorphic encryption paradigms enable strong security and privacy guarantees in an internet voting system, there are still many weak links. For examples, attackers can exploit the poor security guarantees of personal electronic devices if voting is not conducted on public kiosks. By exploiting zero-day vulnerabilities, attackers can install simple key loggers that capture sensitive voting data.

3. Interruption

Although more efficient, internet voting depends on more sophisticated digital hardware and software. This implies that there are more moving parts in the system that attackers can target. With an arsenal of botnets, attackers can conduct Distributed Denial of Service attacks to cripple the normal functioning of the voting system. This, clearly, is much easier to conduct than swarming physical voting centers.

4. Modification

With well implemented end-to-end encryption, the integrity of a voter's input can be assured once it enters the system. However, attackers can still modify the input before it enters a

secure system. For example, attackers can install malwares that have misleading user interface that mimics authentic voting interfaces. Thus, users may be tricked into using an unsafe application, which modifies their votes before submitting to the secure voting system.

Evaluation of Voting Schemes

I personally believe the traditional paper-based voting scheme is far more secure as compared to internet voting.

On a macro level, it is simply much easier to achieve a systematic attack on a digital system as supposed to a human-based system. For instance, exploits like WannaCry may infect hundreds of thousands of computers within a short period. Humans, although sometimes easy to exploit financially and emotional, are not as homogenous in their vulnerabilities. It is, thus, much more difficult to simultaneously convince thousands of voting officials across a country. Unless of course, there is already systematic corruption in the political, and as an extension, the voting system. But the discussion of paper-based or internet voting is hardly relevant at that point.

Furthermore, voting on personal electronic devices is an extremely weak link that undermines the entire system. As we have explored above, many vulnerabilities arise from entrusting a device not specifically designed to ensure strong security guarantees. This could potentially be solved by using an Australian Ballot model, where voters submit ballots privately (into an electronic system), but in a public setting (with publicly provided devices). Then, however, we are once again burdened with ensuring the security guarantees of those terminals.

This issue is further complicated by the fact that we do not have sophisticated trust models when it comes to electronic systems. Whereas humans establish rapport and trust, and can generally have an intuition of who is trustworthy, we find it difficult to ascertain whether a device has been compromised. Even though end-to-end encryption is possible and is unlikely to be crackable in the near future, we can never really be sure of the final result without strong trust enabling paradigms for end terminals.

Question 2: Privacy vs. Security (6 marks)

Importance of Privacy vs. Security

Personally, privacy is more of a concern to me, but I recognize that it may be difficult to enforce privacy without strong security guarantees. For example, I entrust sensitive data of my online behaviours to various social networks. I expect my privacy to be protected by the companies operating these social networks. Without strong confidentiality guarantee in their data infrastructure, it is impossible to maintain the privacy of my social footprints. Thus, I believe governments and companies are justified in spending more money on security, since it indirectly strengthens privacy protection.

Edward Snowden's Whistle Blowing

I personally believe Edward Snowden's actions are justified from the legal perspective of whistle blowing. It can be classified as a security breach of government systems in a

conventional framework, but that is a result of laws and regulations failing to keep up with the pace of technological change. His actions did threaten national security, but it can be argued that in this case, privacy of millions of citizens takes precedence.

I believe that in any complex system, whether political, legal, or social, there must be balancing forces to ensure no significant abuse of power. In this case, Edward Snowden acts as a countering force for potential corruption in intelligence agencies, which benefits both national security and privacy of the citizens in the long run.

However, it's possible that malicious organizations, previously under NSA's surveillance, may now be aware of the monitoring and change their tactics. Thus, it may be too soon to tell the long-term implication of Snowden's leaks on national security, but so far no major harm has come of it.

Question 3: Identifying Different Compromises (6 marks)

a) Privacy

This is compromise of privacy, since the victim in question has lost information self-determination. The drug company has gained unauthorized control of his or her sensitive medical information without his or her consent. In this case, there is no security breach, since the drug company did not launch a malicious attack.

b) Integrity

This case concerns the integrity of the victim's data access, since he or she can no longer be sure of the authenticity of the data he receives. More specifically, the attackers have replaced the data the user wanted (desired websites) with attacker's intended data (advertisement websites).

c) Confidentiality & Privacy

The hacker has comprised the confidentiality of the social network site, and consequently, threatened the users' privacy. Now that the hacker has unauthorized access to user data, users no longer have complete control over the spread and use of their private data.

d) Confidentiality & Privacy

This case can be controversial depending on whether the child's laptop is considered the parent's possession, and whether if the child has right to privacy (from his or her parent) at all. Here, we assume that the answer to both is yes. Then, the parent has compromised the security of the child's device and undermined the confidentiality of the data stored on said device. The child's privacy is violated as a result.

e) Integrity, Confidentiality & Privacy

This is a classic case of breach of security. The hacker has compromised the integrity of the operating system, and provided the victim with a software that he or she did not want. After the hacker receives the password, the confidentiality of the entire device is compromised since the hacker has full control over the sensitive data and programs. The victim no longer has private control over his or her information.

f) Availability

This case affects the availability of the internet and all associated services provided through it for many users. No data is leaked or altered as a result and no user privacy is compromised.

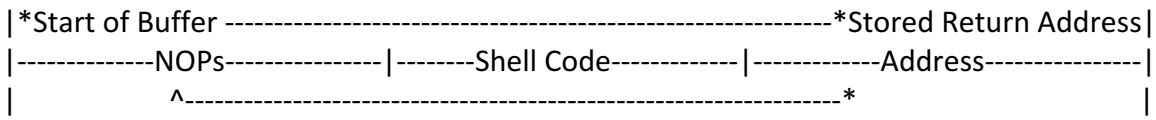
Exploit 1: Buffer Overflow Vulnerability (4 marks)

Description of Vulnerability

Buffer overflow is a class of vulnerability inherent in many of C's input, output, and string manipulation functions. It makes use of the fact that these functions do not check input against a buffer size and will allow attackers to write past the end of the buffer. This implies that attackers can write arbitrary values to the stack past the address of the buffer, including the stored return address.

Explanation of Exploit

The exploit program targets the function copy_file in submit.c, where the entire source file is read to buffer with fgetc without a bound check. To start a root shell, we prepare a long string in the following format:



where the address is calculated to point to somewhere within the NOPs zone.

We write this string to an input file, while paying close attention to aligning "bin/sh" on the boundary of 1024 byte chunks to avoid submit.c's virus detection mechanism. Furthermore, we design the string to be just long enough to overwrite the stored return address, but leave the arguments src_name and dst_name intact so the rest of the function call doest not error out.

After the submit.c executes and calls copy_file, it will return to the somewhere in the NOPs section at the end of the function call. Then, the NOPs are executed until the shell code is reached, where the program will open a shell with root privilege.

Description of Fix

It is extremely simple to fix this vulnerability. One simply needs to enforce a bound check, to make sure the file is shorter than the buffer in copy_file. Or alternatively, the source file can be copied chunk by chunk by restarting at the start of the buffer when the end is reached.