# Critical Infrastructure Security: The Emerging Smart Grid
# Final Take-Home Exam Spring 2012

(Due on May 3)

**Alex Leendertsen, Chao Hu, Cheng-Kuang Yang**

**1.** Mercurial is an excellent, free, distributed source control management (SCM) system, and by far the best SCM system I have ever used. Mercurial can handle projects of any size, no matter the actual size in bytes, or in number of change sets. It also allows an unrestricted number of users to pull and push from any repository, regardless of project (which would be a *really* nasty merge). Since Mercurial is distributed in nature, any clone of any repository is just that, a clone. It can act as if it were the original, or another copy, both of which can be cloned with the same result. Its all the same! This provides unparalleled opportunities for developers at remote location, or developers that develop on multiple versions of projects simultaneously. Mercurial provides all the basic SCM commands like init, commit, push, pull, merge and many more.

Mercurial closely represents the peer-to-peer (P2P) architecture. As stated above, each Mercurial repository acts like an independent node. It can pull and push to any repository/node it wishes, whenever it wishes. Obviously, pulling or pushing to different projects or severely out of date projects will result in an extremely difficult merge, but it is still possible. It is also possible to completely clone any repository/node, and locate it virtually anywhere in a network and have it still function as a SCM system. When a pull takes place, the caller is essentially asking for the differences between his repository and the callee's, and that it be applied to his repository. Push is similar, but in reverse. The caller is telling the callee about changes it has, and to apply them to their repository. Very P2P oriented.

Low Latency Network Connections

Mercurial is not bounded by a network connection unless of course the repository of interest in on a network drive or web server like Kiln. In this case, a low latency connection would be beneficial to developers using Mercurial since it would allow them to perform operations much quicker.

High Bandwidth Availability

As stated above, it would be advantageous for developers to use a low latency/high bandwidth connection when accessing a mercurial repository on a network to provide quick access.

Reply/Answer/Service Consistency

It is very important that the replies from a Mercurial repository are consistent and thus "perfect". If a request to pull changes does not return the correct change sets, the subsequent push could create a new branch sine there are changes in the callee's repository that the caller does not have, or did not get in the first place. The same situation applies to a clone. If not all the data is acquired int he clone, a subsequent push could create unintended consequences. The great part about Mercurial is that it *should* be able to recover from most of these failures, but its better if

they never happen.

The priority of these three is probably the consistency of the response from Mercurial. Users can tolerate high latency/low bandwidth connections, but error prone commands is unacceptable. Thus, the trade off would be bandwidth for robustness. That's not to say that the bandwidth/ latency can get infinitely bad because we all know a slow connection sucks, but making sure Mercurial stays robust is a must.

**2.** Middleware is essentially software that provides services to higher level applications in addition to those provided by the operating system. It can be described as “software glue” since is not part of the operating system, nor part of the software, but provides an interface between the two. Middleware makes it easier for the software developer to achieve basic communication I/O without worrying about its low level intricacies. Middleware in a sense hides the inherent complexity of low level programming and bridges the gap between low level OS communications and programming language simplifications. When a distributed application needs access to a socket, middleware can provide services for naming, location, discovery, replication, protocols, faults, QoS, synchronization, concurrency, storage, access control and authentication.

Much like the way middleware hides low level socket implementations, a high level programming language is an abstraction from the actual low level inner-workings of an OS. High level languages employ more use of a natural languages and are much easier to model and understand than low level assembly languages, making the development of an application much easier.

**3.** Which of the following is false for a NASPInet-like critical infrastructure data delivery service for a power grid:

A. The scale is such that tracking per-flow state at a router (or similar forwarding device) is feasible.
B. The admission control perimeter can be complete.
C. ***Post-error recovery is sufficient.***
D. The predictability of the delivery latency, even in the presence of a small (bounded) number of failures, is very good or better.

**4.** Which of the following is more burdensome for the application programmer:

A. Remote procedure call
B. Publish-subscribe
C. Remote method invocation
D. ***Request-reply protocols***

**5.** Which of the following is space-uncoupled:

A. Remote procedure call
B. ***Publish-subscribe***
C. Remote method invocation
D. Request-reply protocols

**6.** Which is not either a motivating observation or a property provided by P2P systems (i.e., what is false):

A. Compared to exploiting in-network logic, the same level of multi-cast efficiency and very low delivery latency is achievable (even in the presence of a few faults).
B. CPU/GPU and storage resources are often more abundant "at the edges" of a distributed system.
C. ***A much larger address space can be handled compared to IPv4 or even IPv6.***
D. Using only edge resources means that deployment is much simpler than if in-network logic (e.g., GridStat forwarding engines) were required.

**7.** Which of the following challenges for the bulk power grid can not be mitigated with much better wide-area communications (and sensors feeding it):

A. The different physics of some kinds of renewable power (e.g., no reactive power) and how that changes the dynamics of a grid at large.
B. Negligible storage of power in the grid.
C. ***Large numbers of retiring operators who are leaving with a lot of institutional knowledge—intuitive "seat of the pants" understanding—of how their grid operates in many contingencies or unusual situations (and combinations thereof).***
D. Increasing stress on grid due to insufficient new transmission capacity compared to increases in both generation and load.

**8.** Tripwire is a software tool intended to assure integrity of system files by detecting unexpected modifications (such modifications are often a sign of rootkit activity). One version of Tripwire reads the names of the directories to be protected from a configuration file. For each file in the specified directories, Tripwire computes its hash value and stores it in a database. What property must this hash function have?

A. preimage resistance
B. 2nd-preimage resistance
C. collision resistance
D. ***all of the above***

**9.** SYN cookies (s.http://cr.yp.to/syncookiehtml) are used to mitigate SYN flooding attacks.

Which of the following is not used in computing a SYN cookie?

A. Timestamp
B. ***TCP flags***
C. Maximum segment size
D. Port number

**10.** Suppose you have an intrusion detection system detecting a computer virus on the network with 90% accuracy. Precisely, the IDS detects a connection transferring a virus as an attack with 90% probability and a benign connection as an attack with 10% probability. When 1% of the connections contain a virus, what is the probability that a connection flagged by the IDS as an attack is actually benign?

A. ***0–20%***
B. 21–50%
C. 51–90%
D. 91–100%

**11.** The following statements describe Kerberos authentication. Find an incorrect statement.

A. The KDC is a single point of failure.
B. KDC and the ticket-granting server can run on a single machine.
C. Kerberos uses symmetric encryption.
D. ***Every message in Kerberos authentication is encrypted.***

**12.** Which statement is false?

A. Stuxnet exploited multiple zero-day vulnerabilities.
B. Stuxnet attacked systems from certain vendors only.
C. A network physically separated from the Internet can be infiltrated by Stuxnet.
D. ***The author of Stuxnet created two fake certificates to sign drivers.***

**13 (a)** Initial infection

While Duqu and Stuxnet share many similarities, infection is not one of them. Duqu exploits a 0-day kernel attack in which the attacker can use a Microsoft Word document to install Duqu onto the target computer without the user knowing. Stuxnet exploits a vulnerability in Window's shortcut automatic file execution process in order to install itself. When Stuxnet exists on a removable drive and a program like Windows Explorer is used to browse the external drive, the link file would run the copy of the worm. The removable drive will contain a copy itself, and a link to the copy. This is a serious flaw in Windows; some applications that display icons can also inadvertently run code, and in the case of Stuxnet, the code ran in the link file points directly to the actual worm on the same drive.

**(b)** Propagation

Duqu and Stuxnet are somewhat similar when it comes to propagation and replication. Duqu will start by contacting a command and control (C&C) server where it will then install a key logger; it will then record passwords for any devices accessed by the user. Duqu also downloads files from the C&C server that allow it to survey the entire network, pinpointing additional servers and clients. Once enough passwords are compromised, Duqu begins spreading itself across the network. It will begin by copying itself into a shared folder using the authentication provided by the key-logger. Duqu then triggers the copy of itself on the target computer by creating a scheduled task to execute the copy. Another copy of Duqu is now running on the target computer. Instead of connecting to the C&C server again, it connects back to the original computer to receive commands.

Stuxnet also copies itself into network shares, then creating a job to run the copied .dll file. In addition to this method, Stuxnet also uses a few other Windows vulnerabilities in remote procedure calls and a print spooler services to spread itself. The print spooler vulnerability allows for a file to be written to the %System% directory of a target computer, then executing it.

**(c)** Command and control

Stuxnet first tests the internet connection of the infected PC by visiting a website. It then contacts its C&C server using encrypted messages; messages sent from Stuxnet are encrypted using one key, and messages sent from the C&C server are encrypted using another. The initial data sent from Stuxnet contains the Windows version, computer name, network group name, whether or not SCADA software is installed, and IP addresses of all network interfaces. Upon receiving this information, the C&C server can either instruct Stuxnet to execute procedures already existing within its code, or it can give it an additional .dll file to execute.

Duqu does things a little differently. Duqu uses a proprietary protocol that runs on HTTP over port 80 (sometimes with a proxy server), directly over port 443, HTTPS over port 443, or in the case of peer-to-peer (P2P), SMB. Clients can contact other infected computers which will proxy the traffic, much like a P2P network. The protocol itself is a reliable protocol similar to TCP; it even implements fragmentation, reordering, duplicate and missing packets, as well as ACK and sequence numbers.

If the client is using HTTP, it sends repeated GET requests to the C&C server, which replies with attacks to execute in the form of a JPG file. The client can download the new executable attacks and execute them directly in memory or write them to disk. When written to disk, they are encrypted using AES. Files that are saved to disk are decrypted when run.

**14 (a)** *Step 1: Identify all connections to SCADA networks.*

Usually most virus or attacks come from unknown connections, so if we can identify where it linked from, it will be much more easy to prevent attacks from local area and wide area

networks, the Internet, wireless network devices, a modem or dial-up connections and etc.

*Step 2 : Disconnect unnecessary connections to the SCADA network.*

This will create less risks for letting attackers to attack the computer from network, blocking all entrances.

*Step 3: Evaluate and strengthen the security of any remaining connections to the SCADA network.*

Conduct penetration testing or vulnerability analysis of any remaining connections to the SCADA network to evaluate the protection posture associated with these pathways. Evaluating the weak points at every entry will be helpful to prevent such kinds of attacks.

*Step 4: Harden SCADA networks by removing or disabling unnecessary services:*

Since Stuxnet can attack through default network services, disabling unnecessary services reduces the risk of entrance by an attacker.

*Step 5: Do not rely on proprietary protocols to protect your system.*

Do not rely on proprietary protocols or factory default configuration settings to protect your system. Since it doesn't add very much security to your system, best way is to close the back doors or vendor interfaces to your system.

*Step 6: Implement the security features provided by device and system vendors.*

Analyze each SCADA device to determine whether security features are present, and set all security features to provide the maximum level of security.

*Step 7: Establish strong controls over any medium that is used as a backdoor into the SCADA network:*

Disable inbound access and replace it with some type of callback system; Stuxnet can access all the controls to SCADA network and resources.

*Step 8: Implement internal and external intrusion detection systems and establish 24-hour-a-day incident monitoring:*

Alerting network administrators of malicious network activity originating from internal or external sources should prevent attackers to attack the program. Doing this daily prevents attacks in the long term.

*Step 9: Perform technical audits of SCADA devices and networks, and any other connected networks, to identify security concerns:*

The use of these tools will eliminate the entrance that has less protection that an attacker could exploit.

*Step 10: Conduct physical security surveys and assess all remote sites connected to the SCADA network to evaluate their security.*

Identify and assess any source of information including remote telephone/computer network/

fiber optic cables that could be tapped. This can physically cut off attackers from sites.

*Step 11: Establish SCADA "Red Teams" to identify and evaluate possible attack scenarios:*

Feed information resulting from the "Red Team" evaluation into risk a management processes to assess the information and establish appropriate protection strategies. You can also use a variety of people who can provide insight into the weaknesses of the overall network, SCADA systems, physical systems, and security controls.

*Step 12: Clearly define cyber security roles, responsibilities, and authorities for managers, system administrators, and users:*

Establish a cyber security organizational structure that defines roles and responsibilities and clearly identifies how cyber security issues are escalated and who is notified in an emergency can prevent less damage and increase efficiency if an attacker succeeds.

*Step 13: Document network architecture and identify systems that serve critical functions r contain sensitive information that require additional levels of protection:*

Understanding of the functions that the systems perform and the sensitivity of the stored information is vrey important to network defence; such as understanding the overall protection strategy, and identifying single points of failure.

*Step 14: Establish a rigorous, ongoing risk management process.:*

Perform a baseline risk analysis based on a current threat assessment to use for developing a network protection strategy, it will take less damages when getting attack while you have a risk management process.

*Step 15: Establish a network protection strategy based on the principle of defense-in-depth.:*

Defence in depth (also known as deep or elastic defence) is a military strategy; it seeks to delay rather than prevent the advance of an attacker, buying time and causing additional casualties by yielding space. Rather than defeating an attacker with a single, strong defensive line, defence in depth relies on the tendency of an attack to lose momentum over a period of time or as it covers a larger area , and can utilize technical and administrative controls to mitigate threats from identified risks as great a degree as possible at all levels of the network.

*Step 16: Clearly identify cyber security requirements.*

A formal program is essential for establishing a consistent, standards-based approach to cyber security throughout an organization and eliminates sole dependence on individual initiative.

*Step 17: Establish effective configuration management processes.:*

Changes to hardware or software can easily reveal the weakness of the program, either in hardware configurations and software configurations. It is very important to secure the security changes to the process to prevent the attackers from attacking the vulnerabilities.

*Step 18: Conduct routine self-assessments.*

Self-identify issues, conduct root cause analyses, and implement effective corrective actions that address individual and systemic problems. Implement self-assessment processes that are normally part of an effective cyber security program, this way it will be easier to figure out the own vulnerabilities.

*Step 19: Establish system backups and disaster recovery plans.*

You can't say that there's a defense system that is "100%" safe, so you must perform backups and recovery plans just in case, and it will minimize the damage dealt to the system.

*Step 20: Senior organizational leadership should establish expectations for cyber security performance and hold individuals accountable for their performance.:*

Senior leadership should establish a structure for implementation of a cyber security program. This structure will promote consistent implementation and the ability to sustain a strong cyber security program. It is then important for individuals to be held accountable for their performance as it relates to cyber security.

*Step 21: Establish policies and conduct training to minimize the likelihood that organizational personnel will inadvertently disclose sensitive information regarding SCADA system design, operations, or security control:*

This is very important to network security; personnel should be trained strictly to avoid accident and mistake happens, one wrong can cause the system endangered by attackers.

**(b)** This is a very good attempt at preventing attacks like Stuxnet and Duqu. It all depends on how well these steps are followed and implemented, if at all. Since Stuxnet and Duqu use various Windows vulnerabilities that are now known, it is imperative to keep Windows and other applications up to date in an attempt to patch vulnerabilities before they are exploited. Step 6 is also really important: implement the security features provided by device and system vendors. A lot of software and hardware devices already have security built in, we just have to make sure it is enabled and configured correctly. Since Stuxnet and Duqu are already known, it is likely that these devices already protect against these attacks. Step 11 is also an important aspect: establish SCADA "Red Teams" to identify and evaluate possible attack scenarios. This will create a knowledge base about possible new attacks that need protecting against, and can possibly ward off future Stuxnet and Duqu like attacks. Every step is very important and only works if implemented correctly and to its fullest extent. If done so, these should be very effective at preventing attacks like Stuxnet and Duqu.