Guidelines outlining security measures for use of “critical” SW

The EO identifies a desire for guidelines outlining security measures that shall be applied to the federal government’s use of critical software. However, this is simply too broad a topic to be addressed in any reasonable way in any short time frame other than applying the existing vast body of work that has been done over many years worldwide. NIST has the SP 800-53 and related series of special publications that addresses this and continues to evolve over time. There are the ISO standards in this arena, like ISO 27001 and 27002, and other related standards. And there are the Standards of Practice for making governance decisions in this space, that we have published and updated over time, as well as plenty of other related attempts. The problem is not in identifying a list of measures to be applied – the problem is applying them – which is simply not done.

I will take a specific example, that of sound change control, which is essentially ignored, with failure to execute resulting in many large-scale events with serious negative consequences.

Sound change control is not complicated or even all that hard to understand and do. It has been done since at least the 1960s. It involves merely examining changes before invoking them. That is, bothering to check that a change doesn’t screw things up.

The “Solar Winds” attack is a simple example of a lack of change control.

- The update that caused the problems included a change to include an over-the-Internet software package that is under the control of a 3rd party, and which was or became corrupt (not in the accidental sense).
- The problem was as obvious as it can be – a human readable inclusion of a remotely loaded set of software – with no explanation of why this should now be included in tens of thousands of system critical to operations for thousands of companies and many Federal agencies.
- The NSA, responsible for cyber-security for Federal system, apparently never bothered to look. With their classified certainly hundreds of billions of dollars of budget, they apparently don’t check at all for changes in high consequence critical infrastructure provider systems.
- The CIA with its amazing intelligence capabilities never saw it coming, or if they did, the warning wasn’t effective.
- The individual agencies and operators didn’t check it out – they just blindly accepted it.
- The company responsible for the update apparently didn’t have enough of a quality control program to even be aware of all the software they are providing to their customers.
- Nobody, no agency, and no company was punished or fined or held responsible in any way that is externally ascertainable for their failures to do the most rudimentary checks on such high consequence changes.
- And of course nobody else will start doing these things because there is no motivation to do so, given that all of the responsible parties did nothing and were not in any way negatively impacted, other than by the “invisible hand of the market”.

That’s just not good enough. You have to do better - we all have to do better than that.

The solution is to execute on what we already know how to do. Then it comes to setting standards for such execution, existing standards already identify the need to make good decisions, but good decisions are not being made – at multiple levels. Decisions are numerous, the technology doesn’t support it, and the decision-making process is readily undermined. Even if we somehow miraculously set everything up today to be exactly right, a few seconds later, that would no longer be true, because of the dynamics of the systems in use.

Specific questions were identified relating to least privilege, network segmentation, and ‘proper’ configuration. As a starting point, it should be understood that this is about disaggregation of risk (or at least potential consequences), and the specific methods are merely mechanisms to try to achieve this end. At its core, we don’t apply methods for disaggregation of risk very well, or even measure aggregation, and that may be why we look toward these surrogates and methodologies.

Least privilege:

- Sure it sounds like a great idea. Only give things the capabilities (privileges) they require in order to perform the functions they are supposed to perform. But what is the least amount of privilege required in order to do a particular task? That we cannot readily codify. And even if we could, the mechanisms we use in modern system don't segregate privileges to the finest level of granularity, which means we may be able to achieve reduced privilege, but not least privilege. And there are many different ways to associate capabilities to act on and change content and mechanisms, so the concept of least privilege has to be mapped into those association methods. Today there is no mechanism to do this or even a theoretical basis for asserting we can ever do this.

Network segmentation:

- Absolutely. We should 'segment' networks, and there are many different architectures that do this. The selection of architecture is not trivial and no systematic approach has yet been identified and shown to be effective. But at a deeper level, the underpinning of network separation is disaggregation of risk, and unless and until we start to measure consequences (or risks) associated with segmentation architectures and implementations, we will end up segmenting for less than ideal reasons. Perhaps more importantly, segmentation is, essentially always, partial. Networks communicate, whether we want them to or not. The current technology and architecture of the vast majority of commercial off the shelf (COTS) technology intentionally bypasses all attempts to segregate. So segmentation is increasingly a fallacy, even though there is still some level of segmentation feasible for those willing to take the effort to do so. But with technology development and deployment working against segmentation, this is problematic. The best approach today is essentially temporal microzones in which contact it limited to short time periods for limited functions using cryptographic separation, possibly augmented by limited physical separation.

"Proper" configuration:

- Sadly, there is no such thing. That is, the definition of what is 'proper' is not specified and not really specifiable. Proper is contextual to the particulars of the desires/mandates of the entity, and since multiple entities have equities in most systems and mechanisms, they inherently have different versions of 'proper'. Federal systems, for example, have defined configurations for different security requirements, all by rote. But these settings do not effectively trade off the equities of the parties, even within the government itself. They are imposed on contractors and in the best of cases provided mechanistically by automated configuration management systems as part of provisioning and verified by automated audit mechanisms. But these mechanisms are imperfect, even if they could be properly implemented. They are less expensive and time consuming than manual configuration, which is completely infeasible at this time. There are something like 20 protection bits associated with every file and directory on every system even for the low granularity protection models built into most systems. For most systems there are tens of millions of files involved, leading to hundreds of millions of protection bits. Any bit set wrong might cause loss of effective protection for the whole system and recursively to other systems networked together. We also have process protection settings, database protection settings including for each entry in each dataset, network protection settings, network and firewall configurations, anti-malware, -virus, -etc. settings. And so forth.

Guidelines should start with identifying duties to protect and work through risk management before getting to identifying a control architecture, and from there make determinations based on metrics and facts regarding perception, behavior, structure, and content controls such as these. In page 3 we ...[1]

1 Fred Cohen – CEO – fc@all.net – please consider me as a speaker at the workshop