Responsible Cyber Offense

**Dave Aitel**

**January 30,2021**

There are responsible ways to conduct offensive computer operations (OCO), and these overlap heavily with how most offensive organizations view operational security measures - in other words, it is easy to confuse the goal of “not getting caught” with the goal of “being nice to people you hack”. But they are not the same thing, and this paper attempts to derive the beginning of a framework around how offensive techniques should be viewed beyond just at the moment of contact, “left of exploit”, so to speak.

Why would an offensive cyber team want to be “nice”? It’s well understood amongst most players (notably excluding North Korea) that everyone has an interest in avoiding rampant instability. One team’s “persistent engagement” is another team’s unforgivable sovereignty violation. The goals of responsible OCO are not just to avoid collateral damage and unintended conflict escalation or even about compliance with any given legal framework, but to signal to the public and private communities that these capabilities are not too escalatory to maintain and use in a broader sense.

At some level, any paper on responsible offense can be summarized with “be not like NotPetya and Wannacry”. In some cases, policy proposals have been put forward, most obviously [by Microsoft](https://blogs.microsoft.com/on-the-issues/2020/12/17/cyberattacks-cybersecurity-solarwinds-fireeye/) and [private industry groups](https://cyberpeaceinstitute.org/who-we-are), that indicate that essentially all forms of cyber offense are irresponsible and should be curtailed. But protecting the internet from systematic failures is more about producing *predictable effects* than refraining from all exploitation or even avoiding large target sectors (marked inevitably as “critical infrastructure”).

A key takeaway from the analysis in this paper is that it costs more to be responsible[[1]](#footnote-0) in this space than is commonly perceived. Most public cyber policy papers are built around the false idea that cyber operations are intrinsically asymmetric. Yet the costs of a responsible operation are largely driven by *testing*, which is as expensive for sensitive cyber offensive operations as it would be for a large critical infrastructure project. A responsible team also has multiple implants and toolkits in reserve, since the currently deployed infrastructure might need to be torn down at a moment’s notice, multiplying their expenses.

At some level, readers familiar with the area of penetration testing will notice the concepts that guide professional high level penetration teams are the same that should guide offensive teams in the commercial security assessment sector - strict due diligence with targeting, tight scoping, data collection limitations, the ability to respond to your customer, and efforts to limit unpredictable harm. These are all at the root of limiting risk to third parties and the targets themselves.

The responsibilities for an offensive group can be conceptually divided into the following:

* Preparation
* Targeting and Scoping
* Incident Response Response
* Automated Techniques
* Strategy Considerations

**Visible Indicators**

Often a summary chart of these sorts of things is valuable when looking at the indicators of compromise of a particular group, to determine what level of responsibility norms they are adhering to. Groups that share a management structure may also share policies and back-end infrastructure, even if the tooling they use is completely bespoke. The strongest and most useful for differentiation of responsible and irresponsible actors indicators are where there is an obvious **operational cost to implementing responsible practices**.

| **Category of Indicator** | **Visible Toolmark** | **Operational Impact** | **Cost** |
| --- | --- | --- | --- |
| Preparation | Did not port exploit to other versions, platforms, or languages of target, but attempted exploit anyways (discovered when the exploit fails or leaks) | This is usually much better OPSEC, although recon can add to visibility or complexity | Medium |
| Operations | Proper log cleaning and backout procedures when caught | Can be better or worse, depending on the situation. While empty logs are often an IoC, it can be more covert to wipe a disk than try to clean up nicely | Medium |
| Preparation | Exploits and implants tested on proper cyber range - visible when obvious failures occur (c.f. WANNACRY) | Better - although this slows down operational deployment for new tools significantly | High |
| Targeting | Unique toolchains for verticals or for geo-fenced regions | This is better from an OPSEC standpoint, assuming all the toolchains are equally good, but tends to be worse because it means underinvestment in each one | Extremely high - both development and training are expensive |
| Automated Techniques | Worm featuring kill-switch and control mechanisms | This can give defenders an ability to disrupt operations | High |

**Preparation**

Many [academic models](https://www.researchgate.net/publication/272301660_PrEP_A_Framework_for_Malware_and_Cyber_Weapons) for cyber operations include a strict hierarchy for how to analyze the components of the operation that split the exploits and “payload” into separate categories. But it’s easiest to understand that all the parts fit together as a chain, and that the entire chain has an emergent effect of reliability and responsibility.

As a historical example, it used to be that each Windows Operating System came in a number of language flavors, for Chinese, Spanish, English, etc. Each of these flavors had different binaries which had offsets which would change the way an exploit (the many MSRPC exploits of the time, for example) was required to be written. Even though using an exploit only written or tested for the latest Windows Service Pack on the English language would result in an extremely high level of success, [various reconnaissance techniques](https://www.immunityinc.com/downloads/Remote_Language_Detection_in_Immunity_CANVAS.odt) were developed to determine which language of Windows a remote target was running. Yet in some cases, it was impossible to determine the difference between highly related languages (for example, languages that shared a similar character set).

Attackers then had a choice: 1) invest in building a large database of possible offsets to find corresponding offsets that would not crash a target (which required installing and testing against every possible version of Windows in every possible language) or 2) write for the most common language and operating system/service pack levels and hope for the operator to have enough reconnaissance information that they would not use the exploit against a target which might crash.

In some cases, for particular exploits, it was impossible to create an exploit that was generically compatible with multiple languages or service packs, at which point a much higher investment had to be made in finding new ways to do more and more detailed reconnaissance for targeting, and more and more effort had to be made porting the exploit to all of the potential target combinations.

This kind of tradeoff has been seen with the EternalBlue and BlueKeep exploits as well, and is reflected in the public reports of the nomenclature being related to “Eternal Bluescreen” - a comment on the historically poor reliability of the exploits.

Exploit reliability becomes a difficult science to measure, without massive investment in various tooling, seen previously as a set of VMs (and non-VMs because virtualization can change how an exploit performs in ways that will complicate testing), but also with varying network conditions, target configurations, loads and usage patterns, and many more factors which can make an exploit work perfectly well in all lab conditions but not at all in the real world.

As another example, many remote web application attack classes (blind SQLi is the most obvious) affect the underlying databases of their applications, and those databases are often [shared by other](https://twitter.com/WeldPond/status/1350832843539275785?s=20), unrelated applications. It’s important to do the work to know what other systems use those databases before you launch an attack that may temporarily cause extremely high load and DoS those other applications.

As a continual theme, there are a few major policy decisions when it comes to developing and using exploits:

* How much to invest in exploit testing (you don’t have to purchase an Ixia-or-similar setup but you have to [know why you want one](https://www.c4isrnet.com/cyber/2021/01/14/pentagons-weapon-tester-pushes-for-better-assessments-of-offensive-cyber-tools/))
* Where to invest and how much to invest in reconnaissance and target determination
* What to do when potential failure conditions are found while an exploit or implant is in use

In some cases, the value of a potential successful penetration of a target is so high, that many organizations will utilize exploits that have the potential to DoS the target when they fail. Other organizations will fail to invest in proper testing of their exploit under real world conditions, and will therefore find a much higher failure rate in operational use than they have seen in the lab.

This brings us to another point: **a responsible organization will gradually scale in the use of their exploits in real world conditions** so they know they will not have large scale adverse effects. This is because even the best tested exploit cannot properly be understood in terms of its effects on real-world systems. There’s a huge amount of variability and external factors when it comes to exploitation, and modern exploits have to bypass many platform security mitigations. In addition, most cyber ranges are based on large-scale virtualization platforms, and real-world systems running on hard iron with heavier loads, updated from an older history, with multitudes of different non-modelable software stacks (games, for example can make a huge difference since they often come with complicated anti-cheat mechanisms in the kernel) can have different performance characteristics. For these reasons and many others, it’s advisable to test in the real-world slowly.

Implants also require testing at a level that is unusually high even for critical infrastructure software as they must work in conjunction with both currently known software (typically host integrity software will hook the same APIs that an implant hooks) and *all potential future software*. This is not a purely OPSEC decision - while crashes obviously cause detections, they also have a high risk to the target organization’s integrity. Some exploits are simply never going to be reliable enough or able to be cleanly used without side effects.

False flag operations also [carry complications](https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07170728/Guerrero-Saade-Raiu-VB2017.pdf) in this space. When stealing an implant, exploit, or command and control utility from another offensive cyber group, you are also stealing and using another team’s reputation for responsibility when traversing third party networks. This imposes costs upon that organization that they might not have to take should you have used your own or more generic tooling.

Implant encryption is a balance between reliability and confidentiality. Investing in a public key setup with per-target keying is the most secure, but has the potential for being confused by the operators and hence, losing access to targets. Protecting a target’s sensitive information as it exits their network to an attacker’s listening posts is important, since it protects the target’s larger integrity against anyone else listening in with passive [collection](https://www.google.com/url?q=https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/07170728/Guerrero-Saade-Raiu-VB2017.pdf&sa=D&ust=1610563410113000&usg=AOvVaw2OxluDrIQIMcX3AnJmNk6_).

**Targeting**

Separation of operational targeting by geo-fence (i.e. the Middle East) or verticals (Gov/Mil Contractors) allows defensive organizations to focus on the threats most relevant to themselves, and means they don’t have to over-invest in areas of protection that may not be cost-effective. For example, few banks can afford to do full hardware integrity checks on their supply chain, *nor should they have to*. A responsible offensive actor can reserve techniques like this for high priority targets (i.e. governments) which have the capability to absorb the implicated expenses.

Some targets may have an internet or financial posture that would not allow them to recover from an attack, or may already be recovering from an attack. This is not just the risk of downtime from an exploit crashing a server, but also the risk to their reputation or in some cases, their financial condition may not allow them to invest in proper redundancy or modern software stacks. For example, if a responsible team absolutely needs access to CERT or other highly sensitive information, then maybe they will first figure out if there is any other pathway to those observables before targeting the CERT itself.

You can’t put at risk data that you don’t collect, and these are the key principles of making sure you are not weighing heavily on a third party organization. Although from an anti-attribution standpoint, it is sometimes better to collect on a large set of people to obfuscate who the intended targets are, from a responsibility standpoint you want to collect as little data on as few people as possible, while hacking as few machines as possible and installing persistent implants on as few as possible.

Minimization also occurs in time. Implants need a date when they uninstall - lack of one implies an endless scope, which is a sign of irresponsible policy. Likewise, a responsible offensive team will prevent exploit stealing by [encrypting](https://securelist.com/the-mystery-of-the-encrypted-gauss-payload-5/33561/) them [inside their implant](https://securelist.com/the-flame-questions-and-answers/34344/) until use and not using them widely over unencrypted networks.

A responsible OCO organization will MITM as few people as possible. They will remove targets from their attack chain as soon as they can discover they are not their intended target. This is what crimeware often gets wrong - blindly throwing exploits at every target and then deciding in the implant stage if they have valuable information instead of prioritizing as early as possible and erring on the side of not attempting to penetrate a target.

Within a network, responsible operators will endeavour to be [on the most robust parts](https://onedrive.live.com/?authkey=%21AF4G24Oqi%5Fa2LRE&cid=20DED053E77A9001&id=20DED053E77A9001%213914&parId=20DED053E77A9001%213915&o=OneUp) of that network - the parts that don’t need to operate in real time under heavy load or are not single points of failure (domain controllers for example).

**Incident Response Response**

Private organizations must engage in incident response when they discover an ongoing cyber operation, and how an offensive team responds to that response can be telling. From a policy perspective, this may include communicating publicly that the private business was not a target, avoiding areas of the business that are not operationally important, and limiting any affected follow on effects to a minimum number of their customers (for example, during a supply side attack). There is a difference between attacking a select number of SolarWinds customers, and being so careless with your operation that you put SolarWinds out of business.

The P in “Advanced Persistent Threat” does not apply to the tactical activities of a responsible operator when they have been discovered - and in many cases a responsible organization is extremely timid. When they are discovered, they disappear, after cleanly removing their infrastructure or simply shutting it down with a remote kill-flag.

Knowing when you have been caught is a key objective of all cyber operators, but you will also see groups scan their target’s email for upcoming upgrades (from Windows 7 desktops to Windows 10, for example). If they are not able to be confidently reliable in the new environment, a responsible offensive actor will uninstall their implants ahead of the upgrade.

Technologically, it is important that implants can be removed cleanly, that exploits be securely encrypted except when in use on a target endpoint, and that end-to-end encryption is employed to keep any sensitive information from transiting the network in the clear, or leaking to another party doing passive collection.

Backing out is often a more important part of an operation than getting in. Responsible operators need to be able to clean up the lingering flotsam and jetsam of their efforts without deleting the target’s sensitive data. So for example, they will need to carefully [excise their presence](https://en.wikipedia.org/wiki/Winzapper) from the logs, not just by deleting the entire log directory. Often a responsible attacker will assume they *are not the only attacker on the network* and behave accordingly.

In addition to systemic instability, there’s an element of *implied coercion* that is applied to private companies when they have to bear the costs and parasitic load from an ongoing cyber conflict they are really only collateral to. This can be mitigated by responsible OCO organizations during “incident response response” by publishing removal and mitigation information through a friendly (in-country) anti-virus company. In the extreme, sending information to a company’s national CERT, hired incident response firm, or internal security team can help recover from the natural side effects of an operation.

**Automated Techniques**

The future of defensive technology is automated remediation, and hence the future (and much of the present) is automated attack. This is not limited to worms - even manual activities for second stage implant installation and lateral movement are becoming automated, as response organizations push their time to detection shorter and shorter.

Even when run by a human operator, the important question has always been: Can you back out of this network cleanly at every given point in your attack script while protecting both operational security **and the integrity of your target**?

Worms present additional problems for an operator who wants to avoid emergent behavior that is detrimental to the internet as a whole. It is worth remembering that Stuxnet was originally discovered due to unexplainable bluescreens - demonstrating how hard QA can be when implants are running on arbitrary systems.

There are two main fallacies with worms, as used by professional teams, and one is that they are all meant to be discovered, and the other is that they will inevitably be discovered. Worms are not all network-based - many worms exist only within a particular application context, and are spread by scripting injections or even user interaction.

A responsible worm will always have some mechanisms for controlling[[2]](#footnote-1) its spread that are based on the environment it is supposed to be running in, avoid uncontrolled or duplicative replication, have a time-based kill switch, and a file, domain, or mutex kill switch which can be used by defenders once it has been discovered and analyzed. Worms pose the largest systemic risks to a network, and therefore need to “fail open” and have built in rate limiters. It’s easy to assume that some automated techniques that are deployed within air-gapped networks are already self-limited, but this has been proven to be an unreliable measure - self-replicating code eventually gets out, like the dinosaurs in Jurassic Park.

**Strategy**

Much has been made of the idea that in the cyber domain both CNE and CNA look the same to the target. This is true in the sense that deleting data (CNA) is one command different from reading data. However, a responsible strategy for cyber offense **still separates CNA and CNE toolkits, organizations, command and control infrastructure, and targets**. In other words, it is clearly irresponsible to conduct a CNE operation and then turn over access to that target to ransomware operators once nothing else useful is left to be found.

It is entirely reasonable to ask defensive companies to not report on certain efforts when they discover them (counter-terrorism and anti-child-exploitation come to mind), but only if the tooling and target sets for those efforts is unique. This is a confidence-building measure that is built over time. It also, obviously, adds considerable expense.

Separation of toolkits and exploit chains can be complicated by the use of commercially generic (Cobalt Strike) or open source (mimikatz and metasploit) pieces. These offensive tools are often more reliable and tested than a custom-built remote access trojan or password gathering tool. The rule of thumb then, for separation of tooling is not to insist that every portion of the capability be bespoke, but that building separate organizations and technology development strategies will naturally result in differences visible by incident responders.

It goes without saying that when operations are outsourced to a proxy group or contractor, these obligations need to be enforced on that operator.

# Conclusion

Learning how to use offensive techniques responsibly on the wider internet is an adjustment of thought for the community that involves both judging what we know of external operations and our own internal operations and those of our partners. It’s worth engaging in a post incident response review process - how well did your encryption and other efforts work to avoid systemic risk or collateral costs for the third parties you engaged to reach your intended target, and how can you change your own policy to lower those collateral costs without unduly affecting your operational success.

In some sense, the United State’s Vulnerability Equities Process attempted to address this on a very narrow scope - and the concepts that originally drove it are better adapted to a larger field of work.

Having a metric for the responsibility of a particular group, based as much as possible on their observable behavior, can be a useful tool for international relations teams or even incident responders, [similar to the qualitative frameworks](https://cybersecpolitics.blogspot.com/2016/06/useful-fundamental-metrics-for-cyber.html) used for technical sophistication. These can guide our retaliation and escalation efforts, or help us build norms and confidence building measures that reach across adversarial relationships.

1. The Tallinn Manual ([pp30-31](http://www.touchsymposium.org)) for example has a lot to say about the causation of damage, and responsibility that attaches from cyber acts originating from a State or State Proxy. This paper carves out some of the technical and policy areas where these concepts connect to real-world operations. [↑](#footnote-ref-0)
2. There are some specific details on this in “A control measure framework to limit collateral damage and propagation of cyber weapons“ from Raymond, Conti, Cross, Fanelli https://ccdcoe.org/uploads/2018/10/8\_d1r2s6\_raymond.pdf [↑](#footnote-ref-1)