The Order of the Overflow

Zardus — Yan Shoshitaishvili — zardus@gmail.com
crowell — Jeff Crowell — jeff.crowell@gmail.com
adamd — Adam Doupe — adamdoupe@gmail.com

1 Introduction

In the 80s and 90s, the world changed in a fundamental way. One day, it took days to mail a letter, weeks to buy a product, and months in between being able to talk, face-to-face, with loved ones. The next day, emails crossed the globe in seconds, products could be ordered "online" in a matter of days, and a video-call could be had on a whim. But, as with any uncharted domain in human history, there be dragons...

We, hackers, are the dragons of the digital age. For several decades after the interconnectivity of the world began, our civilization did not grasp the concept of security. In the 80s, a clever graduate student disabled the internet by using a "buffer overflow" to inject "shellcode" into a remote process – the first documented (though, certainly not the first actual) use of this technique to take over remote systems and submit them to the will of an attacker. The 90s were filled with tales of the exploits of Kevin Mitnick, Jeff Moss[1], and other "super-hackers", who bent the new laws of digital nature to their will, living as sort of modern Robin Hoods before inevitably ending up in jail and having to go clean. The first decade of the 21st century was marked by constant attacks by "worms", written by hooligan hackers around the world, targeting MSSQL one week, Windows NT the next, and SMB a month later. The internet was the wild west, and these hooligans were cowboys, living and thriving in the lawlessness.

Of course, security improves. The Windows of 2017 is far, far more secure than the Windows of 2007. Vulnerabilities with the impact of that used by the Morris worm (1988) or Slammer (2003) are worth hundreds of thousands of dollars, and are hoarded accordingly. Nowadays, real-world hacking is mostly (though far from completely) concentrated in the hands of nation states, cyber-criminal enterprises, well-trained (and, as a departure from the old Robin Hoods, well-paid) professionals working for corporations, and a few (and far-between) "hacktivists" (and even these have been decreasing in recent years).

Obviously, no one is born with an intrinsic ability to exploit software systems, and there is a gap between noob and pro. Twenty years ago, this gap was filled with script kiddies,

[1] Will flattery help get the proposal accepted?

and the route to transcend beyond that was unclear. Nowadays, the entire spectrum, including complete noobs, those trying to learn, and complete pros, is filled by CTF. CTFs exist that cater to people who don't (yet) know buffer overflows (i.e., PicoCTF), to bright students on their way to security domination (CSAW), and to completely hacking gods, who spend their day-jobs working for secretive corporations and shadowy government organizations.

DEF CON CTF has two roles. One, it provides a venue for the true pro hackers to ply their craft and show off their skill. As such, it acts as a weather wane for the hacking community, pointing out the top hackers (Geohot in his day, Loki in his) and the most effective techniques (tools, automation, etc). But just as importantly, it is a spectacle. Tens of thousands of enthusiasts, noobs, and interested people come through the CTF room. DEF CON CTF is an opportunity to inform, awe, and inspire them so that they will strive to be the next generation of pro hackers and, in turn, inspire others.

The authors of this proposal understand DEF CON. We have collectively been here for a while, from wandering the halls in awe of the master hackers at DEF CON 9 to spending sleepless nights competing against them every year since DEF CON 16 . DEF CON CTF inspired us to rise from our humble beginnings and become relevant contributors and educators to the security field and hacking culture. Thanks to this inspiration, we have created security analysis platforms (angr, radare2, Avatar, and others), hosted CTFs (BostonKeyParty, the iCTF, and BlazeCTF), and educated the next generation of hackers (how2heap, CTF training challenges, and full-on security courses). We have been prominent members of the CTF community for almost a decade, and are excited to lead it into the future.

We propose to shepherd DEFCON CTF, and the overall CTF community, through the era of complete interconnectivity into which our society is entering. Our game design, while maintaining the sandbox free-for-all feel, the elite security challenges, and the rigorous requirements of modern top-of-the-line CTFs, will integrate the concept of multi-stage exploitation through heavily interconnected networks and deployment-dependent asynchronicity. Crucially, our vision includes both gradual change and (in case of unexpected problems, which inevitably arise when hosting a CTF) graceful degradation to time-tested designs. Just as importantly, we will keep CTF a spectacle that can be used to inspire the next generation, who, just like we used to do, will first wander the halls in awe of the players and then hack us to shreds a decade later.

In this document, we will impress upon you our philosophy of what CTF should accomplish, propose a design for the DEF CON 26 CTF, and show that we are the right people to take on this challenge. We look forward to seeing you all at DEF CON 26!

2 The Zen of CTF

The DEF CON CTF is a premier hacking event that benefits at least three target audiences, and any organizer must be aware of them:

The participants. As the elite of the elite in the CTF hacking community, the DEF CON CTF participants deserve a CTF that is fair, is challenging, and pushes them past

their intellectual limits.

The CTF community. As a group of security enthusiasts, amateurs, and professionals, the CTF community dedicates free time and energy to the pursuit of security glory and that knowledge that is obtained along the way.

The spectators. DEF CON brings together folks from all walks of security life, and some of them have never experienced the frenetic energy, joy, and tears of a CTF, and this will be their first introduction.

To properly honor the legacy of DEF CON CTF, and to lead DEF CON CTF into the future, organizers must be cognizant of these different target audiences and to design not only the game, but the stage and room as well, to engage, challenge, and educate these diverse audiences.

2.1 DEF CON as a Leader

Where DEF CON CTF leads, CTF follows. As the "world championship" of the field, DEF CON sets the standard for the challenges, rules, and organization of the CTF. When DEF CON CTF is well-organized, the community has a reference point for its evolution.

DEF CON CTF introduced all of the critical design components of CTF. It popularized the original Capture The Flag concept, introduced the Jeopardy-style challenge board, the "CTF qualifier", and invented both zero-sum and variable CTF scoring.

Therefore, the future organizers of DEF CON CTF must continue this style of innovating in CTF design. DEF CON CTF cannot afford to stagnate and lose its spot as the world championship of CTF. This is critical for both the participants and the CTF community.

2.2 DEF CON as a Proving Ground

DEF CON is where the top teams go to show their skills. The standing of universities, companies, and even countries in the CTF community and beyond is impacted by their presence and performance at DEF CON CTF. Consider DEF CON 25: the absence of Japanese teams from the competitions was well-noticed, as was the fact that there were three Chinese teams. From looking around at the team tables and listening to the accents affecting the whispered IP addresses and memory addresses, one can get a feeling for the health of the security community in a given country. In turn, the presence of a team at DEF CON CTF gives them added reputation to host top competitions in their own countries (for example, the emergence of HITCON CTF as a global force after that team's participation in DEF CON CTF).

Thus, DEF CON CTF must continue as the premiere proving ground for the elite CTF teams. This is important for both the participants and the CTF community. A DEF CON CTF organizer must recognize the importance of the event and have challenges at both quals and finals that reflect.

2.3 DEF CON as a Lens

DEF CON CTF acts as a lens for the entire security community, magnifying the latest vulnerabilities, pushing the bounds of exploitation, and furthering the fields of automated network defense, attack reflection, and fully automated exploitation. The latest and greatest security vulnerabilities make their way into DEF CON CTF challenges. This is an incredibly important part for the community, because reading about a vulnerability description on a blog is not the same thing as actively finding a vulnerability and developing an exploit. There is no knowledge without putting fingers to keyboard, and the DEF CON CTF is perfectly suited to forcing the CTF community to learn about the latest and greatest.

Furthermore, DEF CON CTF pushes people beyond their limits, testing not only a team's binary exploitation skills, but also cryptography, web, mobile, forensics, and many more. This constant pressure keeps teams on their toes—forcing them to keep up with the times and stay relevant and current. There are no resting on laurels at DEF CON CTF.

Thus, a DEF CON CTF must continue to not only having challenging problems, but those challenges must be on the cutting edge of technologies, vulnerabilities, and exploitation. This is important for all: the participants, the CTF community, and the spectators.

2.4 DEF CON as Inspiration

Most importantly, DEF CON is a symbol. It is a statement that hacking is not only cool, not only competitive, not only hard, but also possible and inspiring. Despite their mythical status, these elite hackers are not gods, able to solve problem simply by glancing at them. They make it to DEF CON CTF because they put in the time, effort, blood, sweat, and tear to develop the skills and, more importantly, the knowledge necessary to hack at the highest of levels.

DEF CON CTF must be an inspiration for everyone: the participants, the CTF community, and, most importantly, the spectators. This event should unite everyone: the curious newbie, the grizzled old SOC analyst, the CISO in a suit, the undercover fed, the young students, and the crazy university professor. It is incumbent upon the DEF CON CTF organizers to hold on and maintain this shining symbol of hacker excellence.

2.5 Our Philosophy

DEF CON CTF is a part of our community: a living, evolving event. It needs to be guided, shaped, and shepherded by knowledgeable, careful, and passionate organizers.

Therefore, as future organizers of DEF CON CTF, we hereby promise to uphold and defend the following principles. All of our events, from now to the future, whatever form they may take, will live by these principles.

1. Responsible Innovation. DEF CON CTF must innovate, or it will stagnate and die. However, new additions to the game cannot be added willy nilly. Unlike some lower-ranked CTFs, such as iCTF, DEF CON CTF cannot completely change the game year-to-year and experiment with zany scoring systems or game designs. While

innovation must be pursued for the game to evolve, this innovation must be tempered such that there is no question in the community's collective mind that DEF CON CTF is the world championship of hacking. Therefore, we promise to propel the CTF game into the future—leading the charge—while maintaining the sterling reputation of the DEF CON CTF.

2. Intellectually Rewarding Challenges. Creating a difficult CTF challenge is easy: simply obfuscate or find the vulnerability in a random location in the program. However, this directly contradicts the goals of a top-tier CTF: intellectually rewarding challenges—challenges where you feel accomplished when you solve them, where you had to learn and master a new skill. Our DEF CON CTF will always strive for challenges that are challenging, but in an intellectually rewarding way, not in a random/frustrating way. Never again will participants suffer through a tar ball that creates a qcow file system that contains thousands of deleted files, one of which is a docx that has a comment that contains a bit.ly link to the flag. These types of challenges test dumb brute force skills, and are not the types of challenges that will we have in our DEF CON CTF.

3. State-of-the-art Challenges. Rather than focusing on one class of vulnerabilities or exploits over and over again, we will create challenges have vulnerabilities ripped from both the headlines and the research papers. Cutting edge crypto vulnerabilities that are just theoretical. A massive vulnerability class that topples a Fortune 500. All of these challenges, and many more, will be included in our DEF CON CTF—where theoretical attacks and research blurs the line into the practical and real-world.

4. Inclusivity. DEF CON CTF should be enjoyed by every user. This concept of inclusion extends to the participants, the CTF community, and the spectators. All will feel welcome at our events—all will feel enchanted by the hacker excellence. Young, old, male, female, or person, everyone will be welcome at our event and everyone will feel welcome.

By following these principles, we will deliver a DEF CON CTF that satisfies the needs of the participants, the CTF community, and the spectators.

3 The Team

Our team consists of some of the most experienced CTF players in the world. We have played countless CTFs from close to the very founding of the game. We have also organized 13 CTFs, going back to 2011, of which 6 have been DEF CON CTF pre-qualifying events.

Our leadership consists of three distinguished members of the CTF community.

Zardus (Yan Shoshitaishvili). Zardus has been part of the DEF CON community since DEF CON 9 (2001), part of the Shellphish CTF team since DEF CON 17 (2009), and has been integral in driving Shellphish to qualify for every DEF CON CTF since. In this time, he was involved in hosting 7 editions of the iCTF (with two of

them being DEF CON pre-qualifiers), being specifically in charge of the 2011 iCTF, one of the most innovating and stable iCTFs in history.

In 2011, Zardus became the captain of the Shellphish CTF team, guiding them through the tricky business of being the oldest CTF team in the world until passing on the captain's banner in 2017. As part of this, he has led Shellphish through not only CTFs, but also the creation of tools and training material to benefit the community (such as an easy-to-install distribution of many tools useful for CTF, ctf-tools, and one of the most popular modern references for heap exploitation, how2heap).

Academically, Zardus has led research into groundbreaking binary analysis techniques. In this capacity, he founded, drove, and open-sourced angr, one of the most popular binary analysis frameworks used today. His research has resulted in well over a dozen publications at competitive academic venues, along with multiple talks at DEF CON, Blackhat, HITCON, and RSA. Building on angr, Zardus successfully led Shellphish through the participation in the DARPA Cyber Grand Challenge, in which they won third place and a spot in history.

crowell (Jeff Crowell). crowell is the most enigmatic participants in the modern CTF scene. Playing with Shellphish from 2013, crowell has instrumented many impressive hacks and captured countless flags.

More importantly, he has hosted 5 editions of the Boston Key Party CTF, one of the most respected and consistent competitions out there, of which 4 editions were CTF pre-qualifiers. He has also understood that innovation must be carried out responsibly — while keeping BKP a consistent success, crowell has innovated in the form of an experimental week-long binary-only CTF, BlazeCTF. This sense for responsible innovation is critical to guide CTF into the future. Additionally, crowell has significantly contributed to several editions of CSAW and several other, smaller CTFs.

In his spare time, crowell contributes to the enthusiast community as one of the principal developers of the radare2 binary analysis framework.

adamd (Adam Doupé). Half man, half daemon, adamd brings a fresh perspective to the organizational team by having a deep understanding of both binary analysis, web security, and inclusivity. He has played in 7 editions of the DEF CON CTF, from 2008 through 2013, and has hosted 6 editions of the iCTF from 2010. Critically, he has also studied CTF as a field, producing 4 academic publications about various aspects of the game from the view of an educator.

In 2014, adamd left Shellphish to bootstrap a CTF team from scratch — the ASU pwndevils. This has given him incredible insight into how to best include newcomers in the security field, which will be critical to realizing our philosophical ideals.

In his research, adamd discovered and categorized a new class of web vulnerability: Execution After Redirect (EAR). In 2017, he resumed playing with Shellphish,

quickly becoming the only person on the team able to solve complex (and even simple) web challenge. He hopes to leverage this ability to influence the diversity of the challenges in the game, as we will discuss later in this proposal.

We also have a star-studded cast of CTF players to fill the CTF-hosting trenches.

odo (Sean Ford). odo has played in 10 editions of the DEF CON CTF, since 2008, and hosted the 2008 iCTF. He is a genius at infrastructure, having designed the worldwide, resilient infrastructure for a major payment processing company.

nullptr (Wil Robertson). nullptr has played DEF CON CTF every year since 2004, with a victory in 2005. He is an undisputed master of binary exploitation trickery.

balzaroth (Davide Balzarotti). balzaroth has played DEF CON CTF from 2004 through 2014, and continues to play other CTFs with his students in France. He has astonishingly extensive experience, and many very strongly-held opinions, in binary reverse engineering.

reyammer (Yanick Fratantonio). reyammer has played DEF CON CTF from 2012 through 2018. He has also hosted a number of CTFs, being quite proficient in writing insane reversing challenges and building bulletproof infrastructure.

gsilvis (George Silvis). gsilvis is a cryptographic wizard. While this unfortunately made him less apt to play DEF CON CTF in the past, he has designed a staggering number of crypto challenges for a large amount of CTFs, most significantly BKP.

B-BOT (Brooke Hundtoft). bbot is an organizational mastermind. She has held a number of professional logistical positions, and will ensure that the CTF runs smoothly, both during DEF CON and before it.

Our team is good, but no amount of playing DEF CON CTF and hosting DEF CON qualifiers can prepare one fully to take over DEF CON. A new organizer always has much to learn.

Luckily, in our enlightened society, past organizers of DEF CON retire gracefully, rather than be removed by force. As such, they're not dead, and can be available for help and the occasional guest challenge when they're so inclined. Because we are so lovable, we've managed to "secure" the cooperation of a number of previous DEF CON CTF organizers, who have agreed to offer advice on game mechanics and infrastructure design.

We can do this.

4 Avoiding Conflicts of Interest

Our team consists of currently-active members of the Shellphish CTF team, as well as and formerly-active Shellphish members and friends of the team. This highlights the necessity of avoiding conflicts of interest, as Shellphish intends on continuing to participate in DEF

CON CTF and its pre-qualifying events. Of course, we will take steps to avoid conflicts of interest.

- We will provide no preferential treatment, neither in challenge design (i.e., tailored to the particular skills of the Shellphish team), nor in the selection of pre-qualifying events to favor the Shellphish team, nor the time zone selection for pre-qualification events.
- As the organizers of DEF CON CTF, we pledge store all data on machines completely inaccessible to the Shellphish team. We will not reuse any Shellphish infrastructure or any private code that Shellphish has access to.
- We will not share any non-public information with Shellphish or any other team.
- We will not participate with Shellphish in any event that impacts the standings of teams in DEF CON CTF, such as pre-qualifying events or pre-qualifications for pre-qualifying events.

We have a strong track record of avoiding conflicts of interest — members of our team have run 5 DEF CON pre-qualification events in the last 4 years, and in each of these events, we have successfully segmented the organizing team away from Shellphish. We were transparent to the then-organizers (both DDTEK and LegitBS), and the CTF community as a whole, and would not have proceeded if anyone had objected to our events being pre-qualifications for the DEF CON CTF. While in those events, Shellphish took the extra step of playing under a separate name (usually, `PartOfShellphish`) to easily opt out of the qualifying spot if they had somehow won the event, we expect that Shellphish will continue to play as `Shellphish` in our qualifying event and the various pre-qualifiers, just without us.

In the interest of full transparency, we state this: we are all affiliated with a team which will attempt to qualify for the DEF CON CTF finals that we will host, but we will not favor or provide any special assistance to that team or any other. Our goal is to create a fair and exciting game for all players, and we hope that our reputation as fair CTF organizers speaks for itself in this regard.

5 The Vision

Modern CTF has bug to lag behind modern state-of-the-art exploitation. At pwn2own, Geekpwn, and the like, people combine complex chains of exploits to get from the periphery (i.e., a `<script>`), through the inner walls (such as a sandbox), and right into the heart (for example, the kernel) of software systems. Oftentimes, even this is not enough: hackers are starting to jump from the kernel into hypervisors, Trusted Execution Environments, peripherals (like modem basebands), and so on. This is hacking at a high level, and this needs to be represented in CTF.

Another slice of the spectrum is more broad: the ability to plan and execute complex meta-exploits against remote systems. For example, in 2015, Phineas Fisher hacked into

and practically destroyed `Hacking Team`, using a meticulously planned and carefully executed attack strategy [1]. This level of hacking has also been almost completely unexplored in CTF, and the community would benefit from exposure to it.

In this section, we will detail the steps that we will take to bring CTF in line with the pinnacle of modern offensive security. While the end goal is ambitious, the steps are not large. This is intentional, for two main reasons.

First, the CTF community is fickle and resistant to change, and the leaders have to pull it gently into the future. If done too quickly, confusion, resentment, and backlash reign. One example of this is the attempted introduction of the platform of the DARPA Cyber Grand Challenge, DECREE OS, into DEF CON CTF 2016, which, due to constraints on the autonomous Cyber Reasoning Systems that also participated that year, had to be adopted whole-heartedly in the course of one competition. DECREE introduced many changes very quickly, left hackers "behind" conceptually, and brewed enough resentment to inhibit the adoption of its many good aspects into the CTF community since then.

Second, hosting a CTF is one of the worst things that can happen to you. It is hell. In fact, it is hard to describe just how much of a hell it is. We have hosted literally dozens of CTFs between us, and a CTF with nothing disastrous occurring is a rare CTF, indeed. Thus, it is absolute critical that every change that is made to a proven formula has a graceful fallback for when shit hits the fan (in fact, in CTF, this shit might not just be proverbial).

We have thought about where we want CTF to go, and we have identified the chain of small innovations that will build up to significant change. We will introduce these innovations over the course of our tenure, with an order and frequency that we will continually recheck and reconsider. Some will fail, and we will have ways to recover when this happens. Some will succeed, and, we hope, will be adopted throughout the greater CTF community.

5.1 The Starting Point — Modern CTF

Over the past decade, attack and defense (a/d) CTF has slowly evolved to what we now know as the modern, DEF CON finals style CTF. The concept of the tasks has mainly stayed the course, with the exception of the complexity, while many of the mechanics have evolved. As a representation of the best, most skilled hackers, the difficulty and scoring has moved from favoring armies of relatively unskilled labor, to requiring sharply honed skills in reverse engineering, exploitation techniques, and creativity. Gone are the days of dropping large numbers of with relatively simple to find and exploit tasks, and now we see a slow trickle of complex tasks with a multitude of subtle vulnerabilities that only the elite few deserving of the coveted spots at the CTF table are able to find, exploit, and patch. In order to review the status of a team's own services, traffic from attackers is provided in pcap form. Teams can use the data in the pcaps to understand how other teams are attacking their own services. The modern a/d CTFs consist of approximately 5-8 tasks over the course of 48 hours, divided into rounds of specificied length (order of ten minutes) with unscored nights reserved. For the past 5 years, we've seen scoring move to a zero-sum game. Each team begins with 1/N of the total points, and in each round that a team is

exploited, some of their points are distributed to each team that successfully exploits them. This is done to prevent a runaway in scoring. Additionally, occasionally, some tasks may be added which are more of the traditional "jeopardy" variety, which add points to the pool, which can then be redistributed in the regular a/d rounds. Teams are able to replace their services with patched versions, provided they maintain successful SLA checks. If a SLA check fails for a round, that team will forfeit points for not providing the required service. A recent addition to DEF CON CTF has been the addition of 'open' patches. When a team patches a service, all competors are given access to the patch, allowing others to analyze the patches. This enables teams to write exploits targeting patched services, steal patches, or add backdoors to patches adding an additional meta-game.

5.2 Diversified Service Versions

In the current model, all teams work with the exact same binary for their services. While this ensures fairness, it has two side-effects. First, it turns the entire set of a team's opponents into a single meta-opponent, degrading the team-vs-team component of an attack/defense CTF. Second, it makes the theft of exploits over the wire a fairly trivial procedure, rewarding the replay of network traffic almost as much as the generation of the exploit itself.

We plan to diversify the versions of a service that each individual team is provided, for example, compiling binary services with different memory layouts or even (maybe for a service or two) for different architectures. This will have a number of effects: it will force teams to priority whom to exploit (rather than thoughtlessly spewing exploits across the entire opponent address range), it will stress their ability to write and adapt exploits, and it will force them to understand an exploit that they steal off of the wire before they can replay it. Thus, by diversifying service versions, we will encourage both strategic thinking and the development of quick exploitation skill (or better teamwork to parallelize exploit development).

There are two main risks to this endeavor: one to us as the organizers and one to the players. For us, this represents a large amount of work: to ensure that our diversification of the service does not result in an unexploitable version, a proof of concept exploit will have to be developed for each version. If there are 15 teams in the game, that may represent a significant amount of work. For the players, there is a risk that their version of a service is somehow "easier" to exploit, or someone else's version is harder in some way that is not considered by our POC.

We can mitigate these issues by introducing the change gradually (for example, diversifying only one or two services in the first year) to limit this sort of unexpected impact and by requiring that a team steal flags from at least two or more other teams before the flags will begin being counted to lower the impact of unintended "easy" service versions.

5.3 Modern Tasks: Attack and Defense of the Core

In 2017, we see public attackers focus much more on the kernel. The price for a jailbreak of an iPhone, requiring a kernel exploit, has ballooned to over 1M USD. Kernels are being

built with self-protection, KSPP, grsecurity, anti-patch technologies, and CFE. To go from $ to # is the ultimate goal of any hacker. And yet, still, we rarely see kernel exploitation techniques, or kernel specific bug classes appear in CTF. They're rare in Jeopardy Style, and entirely absent from attack and defense games.

We see bringing these tasks into DEF CON CTF as an important goal; we want to showcase the most interesting targets with the highest payoff for exploitation. The kernel is a very brittle place, we can't derefernce a null pointer without bringing down the whole system. One bad patch, one errant byte can nuke your machine. We plan to leverage virtualization and snapshotting to make the kernel as simple to restart as modern tasks through xinitd are. Players have no experience in defending their kernels, whether by binary patching, or through a new driver that hooks and filters. We will provide players the equivalent of their mobile device with an outdated kernel, or their router from 2011 that still broadcasts as well as it did 5 years ago; a black box that needs wizardry in exploiting to fully control, or the patience in testing to ensure that a painstakingly crafted binary patch won't break usability, all while providing the simplicity of a run of the mill userspace service.

There is also a philosophical problem: large real-world software does not tend to make for good CTF challenges, as there is simply too irrelevant code there that hackers can get lost in. Thus, while a vulnerable kernel driver can work, a vulnerability in a kernel itself will likely result in a frustrating game. One potential solution to this is to use a custom, minimal, CTF-specific kernel, or set of kernel modules for these challenges.

5.4 Multi-stage Services

We plan to bring the modern chaining of vulnerabilities into complex exploits into CTF by introducing services beyond the traditional one-shot exploitable binaries. For example, requiring a player to move laterally through a system in order to reach a deeper target. Modern software like Google Chrome's use of seccomp and brokering, iOS's use of Apple's Seatbelt, and Android's use of SELinux force attackers to escape restrictive software jails in order to attack vulnerabilites found in the system's core. We envision using similar process separation technologies, making CTF competitors hop between userland processes with different capabilities, in order to attack higher privelaged tasks, like kernel modules.

The biggest risk of multi-stage services is that they may be too complex for teams to solve within the 48 hours of the CTF, and teams might de-prioritize them to focus on easier ones. This is not a very convincing fear: DEF CON CTF draws the best hackers, and even with last year's huge curveball in the form of the 27-bit cLEMENCy architecture, competitors were throwing incredible complex exploits and fielding sophisticated patches (even in a partially-automated way) by the end of the competition. Even so, there are mitigations for this risk, introduced in this year's KiwiCTF (and possibly somewhere before that, of course). For its pwnables, Kiwi had two flags: one for causing a segfault, that would be printed by a segfault handler, and one that actually required a successful exploitation. We could adopt a similar approach to multi-stage services, rewarding gradual progress with partial flags.

Of course, this change can also be introduced on a per-service basis to mitigate the risk

of fucking things up.

5.5 Inter-service Connectivity and Multi-path Exploitation

To better reflect today's complex connectivity topologies, we plan to slowly introduce dependencies between services. In the full vision, these would be hard dependencies: for example, the need to compromise a network-facing server before pivoting to a database-analogue and finally penetrating into the kernel of the database machine (more on the risk of this in Section 5.6) to achieve code execution on the machine's CryptoCard and leak the flag from it. However, there is no way all of this won't end up as a garbage fire when we actually run the CTF.

To lower this risk of meltdown, we initially plan to emulate inter-service connectivity through the score system. That is, flags from the "DB" service would be worth less if the "web server" service was not also being compromised at the same time. Gradually, whether throughout one CTF or over the years,"worth less" can morph into "worthless", and the dependency can be established.

Eventually, we do plan to attempt to actually connect services together, though perhaps initially by leveraging the DECREE multi-challenge design (for predictability) and only much later moving into actual connected systems. Once this is done, we can move further to a system design where there can be multiple "paths" of exploitation, and different combinations of peripheral and core services that can be exploited to reach the same flag.

One risk of this aspect is the neutering of entire groups of services due to easily-patchable vulnerabilities in the peripheral services. There are several routes to mitigate this: either stuff the peripheral services so full of vulnerabilities that there is no feasible way to patch them all, disallow patching of peripheral services (by polling for the vulnerability), or, after a timeout, remove the patched peripheral services from the game to allow teams direct access to the meaty interior of the exploitation path.

5.6 To The Core, and Beyond: Cyber-physical Flags — The Dollhouse

Once there is talk of breaking into kernels, the range of what can constitute a flag can increases. The flags can be physical constructs that can be access via a hardware peripheral managed by a kernel driver: something spoken out of a speaker, something displayed on a hidden LCD, etc. While reading a flag from /flag or /dev/console can be exciting, there's another level of excitement when taking physical data. The public's ears perk up when news of malware spying on webcams, or even the when the latest shodan search reveals security camera footage from a crematorium control center. The services, then, can be cyber-physical devices that have access to this physical data. At the culmination of this vision, each player would have their services deployed in something like a dollhouse representing a modern "connected office" or something analogous. This enables a number of game styles and can take several forms:

1. Remote user-space services with direct access to flags. For example, a webcam pointed at the LCD might have a backdoor, a memory-corruption vuln, or a com-

mand injection. Using that, the attacker can leak the flag image, OCR it, and submit it. This is analogous to a service with a flag file.

2. Remote user-space services that must be used to pivot into kernel exploitation or other services. For example, a vulnerable service running on a device does not have the necessary permissions to access the physical flag data, so a further exploit must be used to penetrate into the sensor that has access.

3. As discussed in Section 5.5, flags could be reached through different combinations of devices. For example, a flag displayed on an LCD could be retrieved by breaking into a webcam service using a memory corruption exploit or by taking over a powermeter service through a backdoor and analyzing the power usage to determine the flag image.

The risks here are obvious: to begin with, as discussed in Section 5.5, such complex arrangements have a tendency of melting down. In fact, just the kernel services themselves introduce significant complications, both in terms of offense (i.e., exploits bringing down the entire device rather than simply segfaulting a process) and defense (deploying patches for kernel images onto real devices is not as simple as deploying a patched service binary). These are limitations that can largely be addressed by careful implementation and design: watchdogs to regularly reboot devices, a standardized kernel image installation procedure through a web form (similar to challenge submission in DEF CON CTF 2016 and 2017), and probably a round-robin scheduling of which attackers can target which victims. That being said, we plan to introduce this concept gradually, first with traditional services in the form of emulated kernels, then a single device with a kernel vulnerability (similar to the Badger challenge from DEF CON CTF 2015), then gradually expand that.

Other than the technical problems with handling kernels, handling actual physical flags is complex. In the noisy DEF CON environment, it is unlikely that, for example, a microphone service would actually be able to pick up a flag spoken over a speaker. This might limit the concept only to visual flags, which would significantly reduce the novelty.

If the "Dollhouse" concept can be realized, then having physical flags will open the potential for physical patches (i.e., maybe you can block the view of the camera if the polls don't depend on it).

5.7 Beyond Binary Memory Corruption

The last decade of DEF CON CTF has focused on the exploitation of memory corruption vulnerabilities in binaries. While we plan to keep this as the primary skill required in our game, we will carefully branch out to other types of vulnerabilities. In addition to pushing the boundaries of binary memory corruption exploitation, we would like to explore algorithmic attacks (such as runtime complexity attacks against a security watchdog process), side-channels (such as power analysis, cache timing side-channels, and so on), and cryptographic weaknesses.

We would also like to reintroduce non-binary services into DEF CON CTF. While we do not plan to keep non-binary services a strict minority of the services in the CTF, we

feel that ignoring them altogether not only makes large chunks of modern CTF teams needlessly useless, but also ignores real issues facing the security community today. We are especially interested in the interaction between binary and non-binary services, and the potential for confusion in cross-runtime (i.e., between Python and native, Javascript and native, etc) communication protocols.

Throughout all of this, we will maintain our principle of Responsible Innovation. We will not make these changes quickly, and will not allow them to hamper our game. Instead, we will carefully explore them, targeting for an improved DEF CON CTF for the community.

5.8 Dunk Tanks

Once in a while, it's okay to explore a wacky idea. Imagine this: the contestants walk in on day two to a room with a set of new additions: a dunk tank for every team. Each team selects a champion who will sit in the dunk tank and furiously hack (on the provided, carefully-mounted PC with a wireless keyboard and mouse). The services? Dunk-tank controls, getting hacked step by step.

Hack.
Hack.
Hack.
Splash.
The crowd goes wild!

6 The Game Plan

So, that's our vision. It's a crazy vision, but it is ours. One year, the vision will be realized: teams of hackers will be torturing themselves and each other over a meticulously-constructed dollhouse full of complex, diversified, multi-stage services, connected with a complex topology that allows them to take different paths to recover flags, "cyber-physical" flags representing actual real-world phenomena, and a smooth-running game with no issues to boot. But next year won't be that year. As mentioned earlier, the CTF community needs to be carefully coaxed into the future, and the changes we want to bring to DEF CON CTF will take a number of years of gradual experimentation to achieve.

As in the past, we plan to keep the 8 players at a table, combined with 8 hours of rounds with unscored nights. This is in order to give players a chance to strategize and work on exploits or patches without punshing players who need a rest. Players will show up to the game with nothing more than a laptop and networking equipment, and quickly be transported to our dream-world, and their nightmares of CTF.

Furthermore, qualifiers will continue as they have, online, jeopardy style, with the top elite teams accepted to join us in Vegas for the final game. We love the idea of DEF CON as the world championship of hacking, and will follow in the footsteps of the previous years in selecting the top competitions from around the world as prequalifing events, taking the champions from those games, and giving them a spot at our table. The number of teams from third-party vs our own qualifiers is still to be decided.

We have a plan to achieve our vision, and we will detail it here, but we fully expect this plan to change in reaction to how well the small steps work and, to an extent, how well they are received. So, in the fine tradition of repressive regimes everywhere, our current expectation for the evolution of DEF CON CTF:

Year 1. In the first year, we will prioritize stability above all else. While we have extensive experience in hosting CTFs, it cannot be denied that DEF CON is going to be more complex and carry much higher stakes than anything else out there. Thus, we will host a very recognizable, traditional attack-defense CTF with some subtle innovations: we will introduce mild scoring dependencies between at most a quarter of the services, we will diversify one service with different versions for every team, and one of the challenges will be an emulated kernel/device with an emulated "physical" flag. The subtlety of these changes to the game will allow us to directly apply our existing CTF hosting experience, and minimize risk for the first year of our tenure.

Year 2. In the second year, we will forge actual connections between services: certain flags will be guarded behind services that can only be reached through the exploitation of other services. However, to avoid the pitfalls discussed in Section 5.5, we will have a fail-open failure mode that will allow us to allow players direct access to the interior services. Additionally, two of these exploitation paths will lead to the same flag, to allow us to explore the implications of this design on the game. Finally, we will deploy one actual hardware service running a custom kernel as the final step in one of the exploitation paths, which an actual physical flag (such as an image displayed on a tiny LCD monitor).

Year 3. Using the knowledge gained from year 2, we will make the third year of our CTF tenure an almost-realization of our vision. We will have chains of services, many including kernel or even hardware components, some of them sharing sets of actually-physical flags. We will still, of course, have fallbacks to adapt to game-breaking actions by players and to failures of our own systems, but year 3 will represent the penultimate realization of our vision.

Year 4. Year 4 is the time to shine. We will reveal the full, actual dollhouse — a model structure with physical flags, chains of services involving physical and non-physical components, pivoting and planning requirements, and multiple diverse service and vulnerability types. We plan to enable one or two vulnerabilities that can be physically patched — perhaps an incorrectly-installed hardware component that enables power usage measurement to deduce the contents of the LCD screen displaying the flag. Naturally, the fallbacks will remain as a last resort, prioritizing the stability and viability of the game over the exploration of the neat concept.

We'd like to stress that this is our current vision. This vision will be refined over multiple years of careful exploration and experimentation. We will try really hard to convince ourselves that it is not a good idea. If we fail in that, we will do it, but our priority is the careful, effective, worthwhile shepherding of the CTF community into the future.

7 Financial Practicalities

We go into DEF CON CTF knowing full well the financial requirements in hosting a top tier competition. While some other CTF organizers have accepted funding from third party companies, we feel that taking corporate sponsorship comes at a price. Taking money from a business exchanges money for total creative control, and leaves the organizers indebted to the sponsor. We are confident that we have a complete vision, and are unwilling to exchange monetary convenience for any say or censorship in tasks, recruiting, branding, or shoving software licenses onto the competitors. In fact, the dangers of such an exchange were demonstrated at the 2010 DEF CON CTF, when DDTEK's sponsor suddenly pulled support for hardware, almost completely sabotaging that year's CTF. DEF CON CTF has always been a phenomenal standalone event, handled without relinquishing creative control, and we plan to carry on the trend.

Luckily, we are in a position to make this independence less painful. That is, a large part of our team is made up of professors, and we have specific university funding that is meant for pushing forward the state of "cybersecurity eductation", under which CTF competitions fit. This is not a guess on our part — we have hosted many CTFs over the years with these resource funds. Additionally, we have significant hardware resources, acquired in the course of our research projects, that we can leverage toward the hosting of a successful competition (and which we have used to host other CTFs as well). Worst-case, our personal funds should be more than sufficient to bring our dreams to life, as they have for the CTF events that our non-academic members have hosted.

We have money, it's creative control we need for our game. Who you choose to be around you lets you know who you are. One CTF in exchange for knowing what we've brought the best possible experience to the players? That's a price we can live with.

8 Conclusion

Running DEF CON CTF, while it carries the ultimate prestige for organizers, also is enormous pressure, and demands dedication in hosting. Our group contains the perfect amount of experience in both hosting and playing. We know what makes a ctf fun to play, we know what sort of effort is required in hosting a top tier CTF, both Jeopardy and Attack Defense styles. Having sat at the tables in Vegas, in Taiwan, in Tokyo, in Luxembourg, in Beijing, in Germany, and every on-line game in between for the better part of a decade, we know what has excited us and driven us to continue the pursuit of hacking. Having run pre-qualifing contests that have sent 5 teams to the DEF CON finals in the past 4 years, we know what is required of an organizer to host a fun and solid game. We are bug hunters, reverse engineers, cryptography experts, exploit writers, web security experts, we've written the tools you use to do the same. We are uniquely qualified to run this game, dedicated to the craft, and can't wait to create the battlefield for the upcomming future of CTF.

References

[1] Phineas Fisher. Hack back! a diy guide. http://pastebin.com/raw/0SNSvyjJ.