Casa de ferreiro, o espeto não é de pau: evoluindo uma plataforma segura para competições de segurança

Lorhan Sohaky de Oliveira Duda Kondo, Bruno de Azevedo Mendonça, André de Freitas Smaira, Paulo Matias

Federal University of São Carlos (UFSCar)
Rod. Washington Luís km 235 – 13565-905 – São Carlos – SP – Brazil

{lorhan,bruno.azevedo}@estudante.ufscar.br, {afsmaira,matias}@ufscar.br

Abstract. *Capture-The-Flag (CTF) are information security competitions. Even though they are organized by experts in the field, the platforms used to run the events are subject to vulnerabilities, just like any other software. Although literature has proposed the NIZKCTF (Non-Interactive Zero-Knowledge Capture the Flag) protocol, in which participants submit a zero-knowledge proof that they have the answers to competition challenges, the implementation of this protocol lacks usability requirements which have only been realized with its use over the years. This paper discusses lessons learned and the adaptations to NIZKCTF made by the organizers of the Pwn2Win CTF from 2017 to 2021.*

Resumo. *Capture-The-Flag (CTF) são competições voltadas à área de segurança. Mesmo sendo organizadas por especialistas da área, as plataformas utilizadas para a realização dos eventos estão sujeitas a vulnerabilidades, assim como qualquer outro software. Embora a literatura tenha proposto o protocolo NIZKCTF (Non-Interactive Zero-Knowledge Capture the Flag), em que os participantes enviam provas de conhecimento zero de que possuem as respostas aos desafios da competição, a implementação desse protocolo carece de quesitos de usabilidade que só foram percebidos com sua utilização ao longo dos anos. Este trabalho discute lições aprendidas e adaptações ao NIZKCTF realizadas pelos organizadores do Pwn2Win CTF de 2017 a 2021.*

1. Introdução

Competições de segurança cibernética têm sido muito utilizadas como instrumentos educacionais em instituições de ensino para dirigir esforços de recrutamento, qualificação continuada e promoção da marca de empresas ou até mesmo como forma de lazer para grupos que tratam esse tema como passatempo. Apesar desses eventos poderem ser classificados em diversos tipos, os mais fáceis de organizar e mais comuns são as competições de *Capture-The-Flag* (CTF) no estilo *Jeopardy*.[1] Nesse tipo de evento, os competidores têm como objetivo resolver a maior quantidade possível de desafios no decorrer da competição. Os desafios são divididos em várias categorias, tais como criptografia, exploração de binários, engenharia reversa, exploração de vulnerabilidades *web*, dentre outras. Ao resolver um desafio, o competidor toma conhecimento de uma cadeia de caracteres conhecida como *flag*. A *flag* pode surgir, por exemplo, como resultado da decifração de uma mensagem, como conteúdo de um arquivo lido de um sistema remoto

[1] https://ctf-br.org/sobre

comprometido, como entrada aceita por um programa verificador de *flags* analisado pelos participantes ou como texto colocado em uma página de administração de um aplicativo *web*. Ao obter a *flag*, o participante deve submetê-la para uma plataforma disponibilizada pelos organizadores da competição, a fim de obter a pontuação correspondente à resolução daquele desafio. Geralmente, atualiza-se essa pontuação ao vivo em um placar público, a fim de incentivar a competitividade e a corrida por pontos das demais equipes.

As principais competições de CTF internacionais são atualmente reunidas no site CTFtime,[2] que permite a qualquer interessado cadastrar e promover sua própria competição. As competições nas quais determinada equipe obtiver seus 10 melhores resultados são utilizadas para computar a média ponderada da equipe, utilizada para a ordenação de classificação em um placar geral, que reflete desempenho desta ao longo do ano. Após o término de cada competição, o CTFtime promove uma votação da comunidade, cujo objetivo é determinar o nível de dificuldade e a qualidade da competição. Os votos determinam o peso que a edição seguinte da competição terá no cálculo da média ponderada para o placar geral. Qualquer equipe que tenha participado da competição e obtido pontuação superior a zero pode votar. Para evitar que competições de baixa qualidade (com enunciados incompreensíveis ou desafios insolúveis) ou de pouco interesse (com problemas demasiadamente fáceis, atraindo apenas iniciantes) obtenham um peso injustificavelmente alto, as equipes que tenham figurado entre as 50 mais bem classificadas no placar geral do ano anterior também podem votar, mesmo que não tenham se inscrito ou que tenham obtido pontuação zero na competição.

Em 2016, cadastrou-se no CTFtime pela primeira vez uma competição organizada por brasileiros — o Pwn2Win. Esse marco foi importante pois, apesar de algumas equipes estrangeiras já terem participado anteriormente de competições brasileiras, os enunciados dos desafios geralmente eram disponibilizados apenas em língua portuguesa, e as competições não eram amplamente divulgadas ao público internacional. Além de divulgar o trabalho dos brasileiros no âmbito internacional, esse movimento contribuiu para que equipes brasileiras ficassem a par de competições organizadas por outros países e tomassem iniciativa de participar destas. Após 2016, notou-se um aumento de equipes brasileiras participando de competições listadas no CTFtime [Matias et al. 2018].

O Pwn2Win é atualmente uma competição de relevância internacional. Em 2021, a votação do CTFtime atribuiu ao Pwn2Win o peso de 99,41%,[3] qualificando-a como segunda melhor competição no estilo *Jeopardy* do ano até o momento de escrita deste trabalho, atrás somente do 0CTF, organizado pela Shanghai Jiao Tong University. Em 2020 e em 2021, o Pwn2Win obteve resultados de votação superiores aos das qualificatórias principais do DEF CON CTF, a competição de CTF mais tradicional e antiga que se tem notícia. Todo ano, os organizadores do DEF CON CTF escolhem algumas das principais competições do mundo para servirem como qualificatórias adicionais, convidando o primeiro colocado destas para participar das finais do DEF CON CTF em Las Vegas. Em 2021, o Pwn2Win foi escolhido como uma das quatro dessas qualificatórias.

Este trabalho relata as lições aprendidas pelos organizadores do Pwn2Win mantendo e evoluindo a plataforma de submissão de *flags* no período entre 2017 e 2021. Em 2017, após observar casos de competições depreciadas devido a incidentes de segurança

[2] https://ctftime.org
[3] https://ctftime.org/event/1186

em suas plataformas, o Pwn2Win optou por adotar a plataforma NIZKCTF, que utiliza um protocolo criptográfico para evitar que adversários obtenham *flags* às claras caso a plataforma de submissão seja comprometida [Matias et al. 2018].

Desde a adoção inicial do NIZKCTF, os organizadores do Pwn2Win observaram diversos problemas de usabilidade, tentando saná-los em edições posteriores. Os relatos demonstram que mesmo quando os usuários de um software são profissionais de segurança cibernética, estes podem vir a colocar a usabilidade como prioridade em detrimento de propriedades de segurança. Adequar os quesitos de usabilidade sem abrir mão da segurança pode ser um trabalho árduo que exige diversos ciclos de desenvolvimento, principalmente quando não houve um planejamento prévio para utilizar pesquisas de opinião padronizadas nem demais ferramentas do estado da arte em interação humano-computador, obscurecendo assim os reais problemas e dificuldades enfrentadas pelos usuários.

Este trabalho relata, em ordem cronológica, as diversas tentativas de adequação do protocolo NIZKCTF, contrastando com variações em indicadores coletados a partir de opiniões públicas e questionários anônimos direcionados aos competidores. Primeiramente, os organizadores tentaram manter o protocolo original, adequando somente a usabilidade de sua implementação. Em 2021, optou-se por relaxar algumas propriedades de segurança, modificando o protocolo criptográfico. Com isso, alcançou-se pela primeira vez a ausência de reclamações de usabilidade relacionadas ao emprego do protocolo.

Por fim, o trabalho discute como, mesmo após vários anos de adoção de um protocolo criptográfico, podem surgir maneiras de utilizá-lo fora dos propósitos originais. Classificar essas formas de utilização como ataques ou como emprego legítimo do protocolo pode não ser trivial, mesmo quando tomadas como base as opiniões da comunidade. Este trabalho relata uma instância desse problema observada no NIZKCTF em 2021, concluindo com sugestões de modificação do protocolo para trabalhos futuros.

2. Trabalhos relacionados

Mesmo CTFs relacionados a segurança da informação sendo organizados por profissionais ou até mesmo especialistas, é consenso que nenhum sistema digital é completamente livre de vulnerabilidades, de forma que é de extrema relevância para a comunidade como um todo a divulgação das dificuldades enfrentadas na organização de um evento desse porte (que frequentemente conta com dezenas de milhares de competidores simultâneos) e a opinião dos envolvidos. Com isso em mente, é comum a publicação de artigos relacionados a tais eventos, tanto analisando a usabilidade de uma determinada plataforma quanto relatando as lições aprendidas com a organização desses grandes eventos.

Em 2013 e 2014, respectivamente, foram publicados os artigos relatando lições aprendidas com as organizações do picoCTF,[4] organizado por estudantes da Carnegie Mellon University [Zhang et al. 2013], e do iCTF,[5] inicialmente organizado por um professor da UC Santa Barbara [Vigna et al. 2014].

Em 2017, pesquisadores dos Estados Unidos propuseram uma nova geração de CTFs com maior realismo, melhor custo, acessibilidade e aplicações em educação

[4] https://picoctf.org/
[5] https://ctftime.org/event/175

[Taylor et al. 2017].

Em 2018, foi publicado um manuscrito por integrantes de algumas entidades dos Estados Unidos, incluindo a Universidade de Idaho, descrevendo a linguagem ADLES e o sistema a ela relacionado visando disponibilização de exercícios de segurança cibernética para uso na educação [de Leon et al. 2018].

Desde 2018, a ENISA (Agência de Segurança Cibernética da União Europeia) publica anualmente em seu site[6] um relatório sobre o ECSC (Desafio Europeu de Segurança Cibernética) e as lições aprendidas em decorrência de sua organização. Em 2021 a ENISA publicou um relatório bem completo sobre eventos de CTF, mostrando o estado da arte e práticas comuns nesses eventos.

Em 2019, pesquisadores dos Estados Unidos propuseram uma revolução no visual das plataformas de CTF com o intuito de atrair um público ainda maior para esses eventos e o estudo da área [Senanayake et al. 2019].

Em 2020, foram publicados artigos comparando diversas plataformas de CTF de código aberto quanto às suas funções, configurações e utilidades [Kucek and Leitner 2020, Karagiannis et al. 2020].

3. Metodologia

Ao final de cada edição do Pwn2Win, enviou-se um formulário aos participantes contendo dois campos de texto livre: impressões (*impressions*) e sugestões (*suggestions*). Para a análise dos resultados, os comentários enviados como resposta aos formulários foram juntados aos comentários públicos presentes na página de cada edição do Pwn2Win no CTFtime. Primeiramente, foram separados apenas os que fizessem alguma referência à plataforma. Em seguida, foram manualmente classificados quanto a conter elogios ou críticas à plataforma. Por fim, também manualmente, foram agregados quando tratavam do mesmo assunto, a fim de identificar palavras ou conceitos chave.

Em seguida, os comentários foram utilizados para guiar as prioridades de desenvolvimento e evolução da plataforma para a próxima edição. As seções a seguir apresentam um resumo das opiniões coletadas na edição anterior à do ano indicado, seguidas das ações que foram implementadas como resposta a essas percepções dos usuários.

4. Pontuação dinâmica, isolamento de desafios e NIZKCTF (2017)

Em 2016, o Pwn2Win utilizou uma plataforma customizada desenvolvida em linguagem PHP com MySQL e que não era de código aberto. Essa plataforma era um sistema antigo de quando a competição havia sido concebida, em 2014, como uma pequena competição nacional ocorrendo dentro de um evento beneficente.

Na edição de 2016, participaram 312 equipes que resolveram ao menos um desafio da competição e foram coletados 231 comentários. Destes, foram identificados 6 que faziam referência à plataforma. Três comentários criticavam a infraestrutura ou a plataforma, mas não eram específicos quanto ao problema enfrentado e não forneciam sugestões. Dois comentários elogiavam a estética da plataforma, ao passo que um criticava. Um comentário sugeria adicionar um filtro às notícias, mas sem especificar qual tipo de filtro.

[6] https://www.enisa.europa.eu/publications/

Um comentário sugeria que a escala de pontuação dos desafios fosse mais linear em função da dificuldade dos problemas. Como resposta a essa crítica, adicionou-se o requisito de que a nova plataforma deveria utilizar pontuação dinâmica, em que a quantidade de pontos de um problema decresce de acordo com o número de times que resolveram esse problema. É importante ressaltar que a pontuação dinâmica não fornece incentivos para que equipes resolvam determinados desafios antes das outras, pois quando a pontuação de um desafio decresce, os valores novos são refletidos na pontuação de todas as equipes, inclusive as que haviam resolvido esse desafio anteriormente. A vantagem da pontuação dinâmica é que os organizadores não precisam estimar a dificuldade relativa entre problemas — todos iniciam com a mesma pontuação e, no decorrer da competição, naturalmente ajustam-se a um valor supostamente justo, dado que problemas fáceis são resolvidos por mais times. Uma das primeiras competições a adotar essa abordagem foi a edição de 2017 do Google CTF,[7] cuja fórmula de cálculo de pontos foi adotada pelo Pwn2Win.

Outro comentário fazia referência ao fato de que alguns desafios deveriam ser isolados, fornecendo um ambiente separado para cada equipe. De fato, em alguns tipos de problema, a disputa por recursos computacionais compartilhados pode tornar difícil a resolução caso vários participantes tentem explorar vulnerabilidades ao mesmo tempo. Como resposta a essa sugestão, a edição de 2017 provisionou ambientes isolados para cada equipe, acessíveis por VPN, para alguns dos desafios. Como não existiam recursos computacionais suficientes para fazer isso para todas as equipes inscritas, utilizou-se um provisionador automático [Magalhães et al. 2017] que construía os ambientes apenas para as equipes que resolvessem pelo menos 8 dos desafios que não eram isolados e, portanto, estavam disponíveis para todas as equipes desde o início da competição.

Por fim, um comentário fazia referência ao fato que as páginas da plataforma não atualizavam sozinhas. A ausência de atualização automática do placar e de outras páginas da plataforma era intencional na plataforma utilizada em 2016, uma vez que os organizadores sabiam que se tratava de uma plataforma mal dimensionada para competições de larga escala. Mesmo sem atualização automática, os organizadores notaram que as páginas passaram a apresentar lentidão inaceitável nos minutos finais da competição, muito embora essa lentidão não tenha sido citada pelos participantes nos comentários.

Para resolver os problemas de escalabilidade e, simultaneamente, lidar com preocupações de segurança, uma vez que algumas competições haviam à época sofrido ataques que levaram ao vazamento de *flags* ou à manipulação de placar, os organizadores optaram por migrar para o protocolo NIZKCTF proposto por [Matias et al. 2018].

O protocolo NIZKCTF consiste em apenas dois tipos de comando que um cliente, operado por um participante da competição, pode requisitar a um servidor, gerenciado pelos organizadores da competição:

- **Cadastro de time**: O cliente gera um par de chaves Ed25519 pública e privada (pk_t, sk_t) e, em seguida, envia pk_t e o nome do time t para o servidor por meio de um canal protegido por TLS ou SSH (em que a identidade do servidor é atestada por um certificado ou chave pública pinada).
- **Submissão de *flag***: O cliente computa um par de chaves Ed25519 $(pk_c, sk_c) = \mathrm{KeyPair}(\mathrm{PBKDF}(\phi_c, f_c))$, onde KeyPair é um gerador de par de chaves deter-

[7] https://ctftime.org/event/455/

minístico que opera a partir de uma semente, PBKDF é uma função de derivação de chaves baseada em senhas (como scrypt ou Argon2), ϕ_c é um *salt* aleatório e público associado ao desafio (*challenge*) que está sendo submetido e f_c é a *flag* que comprova a resolução do desafio. O cliente verifica se pk_c corresponde à chave pública do desafio e, caso contrário, aborta a operação e avisa que a *flag* está incorreta. Por fim, o cliente gera uma prova $\sigma = \mathrm{Sign}(sk_c, \mathrm{Sign}(sk_t, c))$, ou seja, assina o identificador c do desafio com a chave privada de time e assina o resultado dessa operação com a chave privada de desafio. O cliente envia (c, t, σ) para o servidor, que verifica se ambas as assinaturas são válidas e, em caso afirmativo, publica (c, t, σ) em uma trilha pública de auditoria e atualiza o placar para atribuir a pontuação do desafio c ao time t.

Como trilha de auditoria, a implementação original[8] do NIZKCTF utiliza um repositório Git hospedado nos serviços GitHub ou GitLab e configurado para não aceitar *force pushes*, de forma a evitar que o histórico de alterações do repositório seja reescrito. Como canal de comunicação, utiliza-se o próprio protocolo Git (que pode operar sobre SSH ou sobre HTTP+TLS). As tuplas (c, t, σ) são enviadas na forma de registros de alteração (*commits*) a um *fork* do repositório da competição de propriedade do participante e, em seguida, são submetidas como requisições de integração (*pull requests*) à linha principal do repositório oficial da competição. Um robô de integração contínua atua na figura de servidor, aceitando *pull requests* que validem corretamente de acordo com as regras do protocolo e atualizando o placar da competição.

Como essa implementação usa plataformas de grande porte (GitHub ou GitLab) para hospedar os dados consultados pelos participantes, e como o endereço IP do robô de integração contínua não era divulgado ao público, esperava-se obter capacidade de atender a um grande número de acessos e até mesmo resiliência a ataques de negação de serviço.

5. Melhorias no cliente de linha de comando (2018 a 2019)

Na edição de 2017, participaram 207 equipes que resolveram ao menos um desafio da competição. A redução de equipes participantes com relação à edição anterior foi atribuída pelos organizadores à necessidade de se instalar um software cliente para participar da competição. Foram coletados 55 comentários dos participantes, dentre os quais foram identificados 12 que faziam referência à plataforma. Dois elogiavam a plataforma sem nenhuma ressalva, ao passo que um solicitava que qualquer utilização do GitHub ou de cliente customizado fosse abolida. Um comentário afirmava que a plataforma se sobressaía (*stands out*), mas não ficou claro para os organizadores se isso era um elogio ou uma crítica. Dois comentários diziam que a plataforma precisava de melhorias de experiência de usuário ou de interface com o usuário, mas não eram específicos nem forneciam sugestões. Um comentário dizia que o isolamento de problemas e fornecimento de VPN era desnecessário. Por fim, um comentário sugeria tornar mais fácil compartilhar uma mesma chave SSH para acesso ao GitHub entre diversos integrantes de uma mesma equipe, e quatro comentários sugeriam tornar mais fácil a instalação da plataforma.

Os organizadores entenderam que havia insatisfação com relação à plataforma, mas a maior parte das críticas parecia estar relacionada à dificuldade de instalação ou

[8] https://github.com/pwn2winctf/2017

adaptação da plataforma para necessidades específicas de determinado ambiente (por exemplo, mapear uma chave SSH de um diretório fora do padrão). Para resolver esses problemas, o cliente NIZKCTF da edição de 2018 foi empacotado nas opções de contêiner Docker e de contêiner LXC.[9]

A edição de 2018 teve um decréscimo para 163 times que resolveram ao menos um desafio da competição, provavelmente devido a um conflito de datas que ocorreu com outra competição importante. Assim, foram recebidos também menos comentários de participantes. De 29 comentários dessa edição, foram identificados 5 que faziam referência à plataforma. Desses, um era um elogio sem ressalvas. Dois criticavam a estabilidade da VPN ou diziam que alguns dos desafios que foram colocados atrás da VPN não precisariam estar em ambientes isolados. Por fim, dois comentários sugeriam que a interface com o usuário deveria ser mais amigável, e um que ela deveria ser baseada em *web*.

Devido ao aumento de comentários negativos com relação ao uso de VPN e isolamento de alguns dos desafios, os organizadores optaram por não utilizar mais esse tipo de infraestrutura a partir da edição de 2019. Em vez disso, passaram a focar em tentar tornar o isolamento desnecessário e, quando não era possível, passaram a utilizar filas para dar acesso a um participante por vez aos serviços, utilizando mecanismos de prova de trabalho como o Hashcash [Back 2002] para evitar que a fila fosse monopolizada por um único participante.

Com relação à plataforma para submissão de *flags*, considerou-se pela primeira vez a possibilidade de implementar um cliente *web* para o protocolo NIZKCTF. No entanto, não foi encontrado tempo hábil para realizar essa implementação. Com isso, os organizadores restringiram-se a implementar um *script* denominado `dctf` para facilitar o uso do contêiner Docker, a atualizar e melhorar a documentação e a melhorar a redação das mensagens de erro.[10]

6. Construção de um cliente baseado em *web* (2020)

A edição de 2019 contou com 220 equipes que resolveram ao menos um desafio, ou seja, foi a primeira vez que o número de participantes subiu com relação à edição anterior desde 2016. Antes do término dessa edição, os organizadores anunciaram que tinham planos de desenvolver uma plataforma *web* compatível com o NIZKCTF para a edição do ano seguinte. Dentre 26 comentários coletados, foram identificados 6 que faziam referência à plataforma. Três elogiavam o uso de Git pela plataforma, mas diziam que ela ainda era difícil de usar ou que ainda faltava documentação. Dois diziam que não fazia sentido usar o GitHub e sugeriam que o Pwn2Win passasse a usar o CTFd,[11] plataforma que é utilizada em grande parte das competições de CTF, mas que não fornece as propriedades de segurança do NIZKCTF. Por fim, uma pessoa dizia que não era fã da plataforma, mas que confiava que ficaria melhor ano seguinte (dado que seria uma plataforma *web*).

Apesar dos comentários sugerindo abolir o uso de Git, havia uma quantidade maior de pessoas que elogiavam esse mesmo aspecto da plataforma. Desta forma, os organizadores entenderam que a principal queixa dos usuários era, ainda, com relação ao uso do cliente por linha de comando.

[9] https://github.com/pwn2winctf/2018
[10] https://github.com/pwn2winctf/2019
[11] https://ctfd.io

Assim, para a edição de 2020, foi implementada uma versão *web* do cliente NIZKCTF totalmente compatível com a primeira versão do protocolo.[12] Essa foi uma mudança bastante impactante, uma vez que os jogadores não precisavam mais instalar um programa para participar do evento. A partir desse momento, todo o processo de prova de resolução de desafio e cadastro do time foi realizado pelo navegador, ou seja, os processos criptográficos, autenticação com o GitHub e todas as interações com GitHub para criação de *forks*, *pull requests* e *commits* eram realizadas por meio do navegador do participante.

A interface gráfica foi desenvolvida em Vue. Para criptografia, foi utilizada a libsodium compilada para WebAssembly. A autenticação com o GitHub foi feita por meio do protocolo OAuth. Os *commits* no repositório de submissões foram implementados por meio de uma API específica para essa tarefa disponibilizada pelo GitHub, em vez de usar diretamente o protocolo Git. O cliente de linha de comando já realizava as demais tarefas relacionadas ao repositório Git por meio da API do GitHub, portanto estas foram puramente transcritas para JavaScript. Como a API do GitHub tem restrições de *Cross-Origin Resource Sharing* (CORS),[13] os organizadores da competição tiveram de hospedar um pequeno servidor intermediário (*proxy*) para o GitHub, a fim de adicionar cabeçalhos permitindo a realização de CORS para o site da plataforma.

7. Autenticação por senha e eliminação do repositório Git (2021)

Na edição de 2020, participaram 401 equipes que resolveram ao menos um desafio da competição. Foram coletados 60 comentários dos participantes, dentre os quais foram identificados 11 que faziam referência à plataforma. Um elogiava a plataforma sem ressalvas, e afirmava ser bom existir algo diferente do CTFd. Duas pessoas afirmavam não gostar da plataforma, sem especificar o motivo ou fornecer sugestões. Oito comentários criticavam o uso do GitHub.

Observou-se, assim, desconforto com relação à autenticação por meio do GitHub. Grande parte das reclamações era relacionada à granularidade de permissões disponibilizada pelo serviço. A plataforma tinha acesso a outros repositórios públicos dos jogadores, ou seja, não existia granularidade para garantir que projetos armazenados em outros repositórios públicos não seriam sabotados. Apesar do código fonte da plataforma ser aberto, os competidores não quiseram auditá-lo para ter certeza de que esta não causaria alterações em seus outros repositórios e tampouco podiam ter certeza de que o código que estava sendo executado durante a competição era de fato o disponibilizado publicamente. A necessidade de usar um *proxy* CORS também gerou em alguns participantes um receio com relação à possibilidade de roubo do *token* OAuth por terceiros que viessem a comprometer esse *proxy*.

Além disso, notou-se que alguns participantes preferem permanecer no anonimato, sem ter que estabelecer uma ligação entre sua participação na competição e suas atividades profissionais cotidianas no GitHub ou GitLab, nem criar uma conta apenas para participar da competição, o que é contraindicado pelos próprios serviços.

Reagindo a essas reclamações por parte dos competidores, em 2021 foi implementada uma terceira versão ainda executada a partir do navegador,[14] mas em que a ne-

[12] https://github.com/pwn2winctf/NIZKCTF-js/tree/v2.1.1
[13] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
[14] https://github.com/pwn2winctf/nizkctf-v2

cessidade de conta Git foi substituída por autenticação comum com e-mail e senha implementada por meio do serviço de autenticação do Firebase. Além disso, substituiu-se a necessidade de compartilhar uma chave privada entre os membros do time (para identificar quem faz parte de qual time) pelo uso de uma única conta por time. Os organizadores tomaram essa decisão pois era comum que alguns times registrassem-se com muita antecedência e perdessem sua chave privada, ou que tivessem dificuldade para compartilhá-la entre os membros, gerando demanda para a equipe de suporte da competição.

Assim, o protocolo foi simplificado para o seguinte conjunto de operações:

- **Cadastro de time**: O cliente, previamente autenticado por e-mail e senha, envia o nome do time t para o servidor por meio de um canal protegido por TLS (em que a identidade do servidor é atestada por um certificado).
- **Submissão de *flag***: O cliente computa um par de chaves Ed25519 $(pk_c, sk_c) = \mathrm{KeyPair}(\mathrm{PBKDF}(\phi_c, f_c))$, onde $\mathrm{KeyPair}$ é um gerador de par de chaves determinístico que opera a partir de uma semente, PBKDF é uma função de derivação de chaves baseada em senhas (como scrypt ou Argon2), ϕ_c é um *salt* aleatório e público associado ao desafio (*challenge*) que está sendo submetido e f_c é a *flag* que comprova a resolução do desafio. O cliente verifica se pk_c corresponde à chave pública do desafio e, caso contrário, aborta a operação e avisa que a *flag* está incorreta. Por fim, o cliente gera uma prova $\sigma = \mathrm{Sign}(sk_c, t)$, ou seja, assina o identificador t do time com a chave privada de desafio. O cliente envia (c, t, σ) para o servidor, que verifica se o usuário está autenticado como membro do time t e se a assinatura em σ é válida e, em caso afirmativo, publica (c, t, σ) em uma trilha pública de auditoria e atualiza o placar para atribuir a pontuação do desafio c ao time t.

A proposta original do NIZKCTF protege contra um adversário que compromete o software da plataforma e, em seguida, envia uma resposta (de conhecimento do adversário) em nome de diversos participantes, pois mantém a chave privada do time apenas do lado do cliente, e assina todas as submissões com essa chave. No entanto, isso gera o inconveniente do participante ter que gerenciar manualmente essa chave, copiando-a para colegas de equipe ou para outros computadores que deseje utilizar no decorrer da competição. Argumentamos que, caso o adversário conheça a resposta de um desafio, ele pode simplesmente publicar essa resposta na internet, por exemplo na sala de bate-papo da própria competição, atingindo o mesmo objetivo que o NIZKCTF propõe-se a evitar, porém sem a necessidade de comprometer o software da plataforma. Dada a impossibilidade de proteger contra esse ataque mais simples, pode-se desconsiderar o ataque mais complexo que NIZKCTF propõe-se a evitar, dispensando a necessidade da chave de time.

Com a remoção do Git, tornou-se necessário armazenar os dados da competição em algum outro meio. Para isso, optou-se por utilizar uma base de dados tradicional, em PostgreSQL. Embora possa parecer que o requisito de um terceiro confiável para a preservação do histórico tenha sido quebrado, apenas houve uma adaptação para um programa externo[15] que periodicamente coleta dados da plataforma, realiza uma conferência do placar com a trilha de auditoria e, em seguida, salva uma instantânea da trilha de auditoria e do placar em um repositório no GitHub, atestando os *commits* por meio de registro de provas de existência em um *blockchain* público [Mendonça and Matias 2021].

[15] https://github.com/pwn2winctf/nizkctf-admin-toolkit

A existência de um software de auditoria externo permitiu que uma falha de implementação do servidor *web* que causava ordenação incorreta do placar fosse identificada e corrigida logo no início da competição, antes mesmo que o suporte fosse acionado pelos participantes. Como o placar apresentado pela plataforma *web* não batia com o computado pela ferramenta de auditoria, os organizadores da competição receberam alertas e notificaram os desenvolvedores para correção da falha.

Com relação aos detalhes de implementação, houve a alteração do *framework* utilizado para o desenvolvimento da interface gráfica de Vue para NextJS, a fim de tentar obter páginas com menor tempo de carregamento, por meio do uso de dados pré-carregados. O servidor foi inteiramente implementado em JavaScript, a fim de promover uniformidade de linguagens de programação entre cliente e servidor. O servidor foi colocado atrás da *Content Delivery Network* (CDN) da Cloudflare, e as políticas de *cache* de todos os *endpoints* foram revisadas para que o *cache* da CDN obtivesse alto índice de acerto (74%).

A edição de 2021 contou com a participação de 720 equipes que resolveram ao menos um desafio. Foram coletados 35 comentários, dentre os quais 5 faziam referência à plataforma. Dois reclamaram de lentidão na atualização das páginas, que ocorreu devido a políticas do lado do cliente implementadas pelo NextJS. Dois relataram a existência de *bugs* na plataforma, mas não mencionaram quais, muito embora provavelmente estivessem se referindo às falhas de ordenação do placar observadas e corrigidas logo no início da competição. Um sugeriu que uma tabela com os problemas resolvidos fosse exibida na mesma página do placar.

8. Síntese e discussão dos resultados

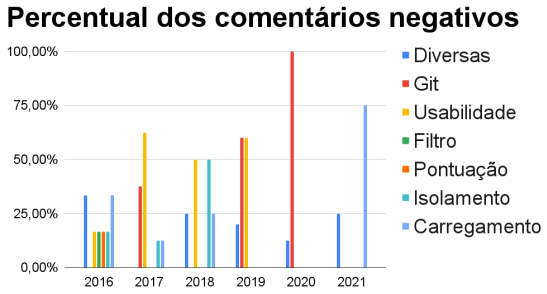

Figura 1. Histórico de críticas por assunto

Detalhada ano a ano a evolução da plataforma utilizada no Pwn2Win CTF, pode-se agora analisar o resultado dessas alterações ao longo dos anos, por meio de um gráfico mostrando a incidência de assuntos nos comentários. O gráfico da Figura 1, que exibe o percentual dos comentários negativos em que aparecem cada um dos assuntos destacados, mostra como a plataforma melhorou na visão dos competidores em tópicos específicos (filtro, pontuação, isolamento e carregamento), especialmente depois de 2018, quando se retirou completamente o uso de ambientes isolados. Em relação ao carregamento, a melhora se manteve na verdade somente nos anos intermediários, mas o problema voltou a ser muito relatado em 2021, e será o próximo desafio para os organizadores em 2022.

Já o uso do GitHub e a usabilidade da plataforma (assuntos relacionados) tiveram um grande aumento nas reclamações até o ano de 2020, enquanto tentava-se contornar os

problemas de forma puntual sem prejudicar o uso do protocolo, mas ainda mantendo a necessidade do uso do GitHub pelos competidores, o que fez com que as críticas em relação a esse serviço aumentassem ao longo do tempo, chegando a 100% das reclamações relacionadas à plataforma em 2020.

Em 2021, quando foi retirada definitivamente qualquer interação direta entre os competidores e o GitHub e a plataforma foi inteiramente baseada em *web*, reclamações quanto à dificuldade de uso da plataforma ou ao GitHub em si cessaram completamente, restando apenas críticas quanto à eficiência da plataforma e considerações puntuais diversas relacionadas a opiniões individuais de alguns jogadores, estas presentes em todas as edições. Essa melhora na percepção refletiu-se na pontuação do evento no CTFTime, a maior da história da competição e a segunda maior dentre todos os CTFs realizados em 2021 até o momento da escrita deste artigo: 99,41%.

9. Açambarcamento de *flags*

Açambarcamento de *flags* (no inglês, *flag hoarding*) é a prática de esconder as *flags* que são encontradas ao resolver os desafios, submetendo-as apenas próximo ao final da competição. Algumas equipes usam essa tática para deixar o primeiro colocado confiante de que ninguém vai passá-lo no placar, na esperança de que este reduza os esforços depreendidos na resolução de um número maior de desafios.

Na edição de 2021, a equipe DiceGang[16] utilizou a tática de açambarcamento contra a equipe uuunderflow, que até próximo do término da competição, encontrava-se na primeira colocação, como ilustrado na Figura 2.

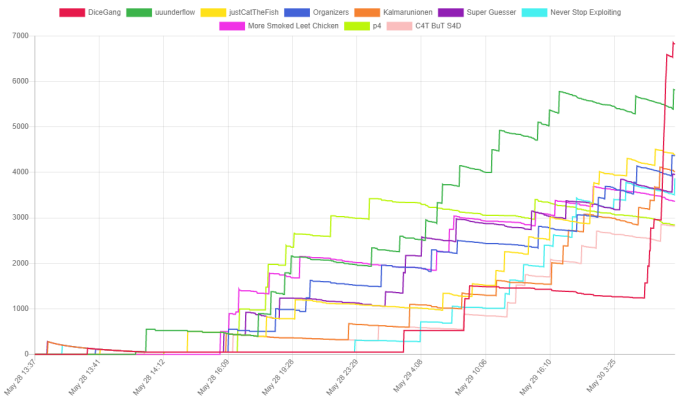

Figura 2. Evolução dos 10 melhores colocados no Pwn2Win 2021

Muito embora essa prática não tenha sido mencionada nos comentários coletados de participantes, posteriormente a equipe DiceGang publicou, em conjunto com a resolução de um dos desafios,[17] uma nota informando que eles haviam feito uso de propriedades do NIZKCTF para praticar o açambarcamento com maior segurança. Como o protocolo permite validação *offline* das *flags*, devido à chave pública dos desafios ser divulgada logo no início da competição, participantes que desejam praticar o açambarcamento podem validar as *flags* com antecedência sem submetê-las, adquirindo a certeza de que elas serão aceitas quando do final da competição.

[16] https://ctftime.org/team/109452
[17] https://ctf.harrisongreen.me/2021/pwn2win/highest_power

A validação *offline* de *flags* originalmente havia sido introduzida propositalmente no protocolo NIZKCTF. O objetivo era possibilitar que as equipes conseguissem continuar participando do CTF caso a plataforma de submissão saísse do ar pois, apesar de não ser possível submeter as *flags* imediatamente, seria possível ao menos validá-las.

Para entender melhor o quanto a prática de açambarcamento era enxergada como um problema pela comunidade, e se as vantagens proporcionadas pela validação *offline* eram percebidas como superiores aos riscos da prática do açambarcamento, os organizadores enviaram um formulário de múltipla escolha anônimo aos e-mails de contato cadastrados pelas 20 equipes mais bem colocadas no Pwn2Win 2021.

Como resposta à pergunta "como seu time se sente sobre açambarcamento de *flags*?", 28% responderam que "faz parte do jogo"; 36% responderam que "é uma conduta antidesportiva, mas os organizadores do CTF não devem interferir"; e 36% responderam que "é uma conduta antidesportiva, e os organizadores do CTF devem tentar coibir". Desta forma, percebe-se que se trata de prática polêmica, que a maioria considera como antidesportiva e que uma parcela significativa dos participantes acredita que os organizadores deveriam tentar coibir.

Como resposta à pergunta "das *flags* que seu time encontrou durante o Pwn2Win 2021, alguma resultou em erro de validação (ou seja, você encontrou uma *flag* falsa ou inválida)?", 8% responderam "sim"; 52% responderam "não"; e 40% responderam "não lembro / não sei". No entanto, a baixa incidência de erros de validação pode ser utilizada para argumentar tanto a favor quanto contra a validação *offline*. Por um lado, ela demonstra que times que desejem fazer açambarcamento ganham pouco conhecimento adicional devido à possibilidade de validação *offline*, pois quaisquer *flags* encontradas provavelmente serão verdadeiras. Por outro lado, argumenta-se que como a validação *offline* não é necessária para ter certeza de quaisquer *flags* encontradas, ela não deveria ser permitida, pelo princípio do mínimo privilégio.

Como resposta à pergunta "os benefícios da validação de *flags* do lado do cliente sobrepujam os riscos relacionados ao açambarcamento?", 12% responderam que "concordam fortemente"; 16% que "concordam"; 32% que "não concordam nem discordam"; 20% que "discordam"; e 20% que "discordam fortemente". Desta forma, percebe-se que a maioria dos participantes da pesquisa acredita que a plataforma não deveria permitir a validação *offline*.

A fim de remover a possibilidade de validação *offline* e preservar as demais propriedades de segurança do protocolo, o NIZKCTF poderia ser alterado para usar o esquema de assinaturas Ed25519-MuSig [Maxwell et al. 2019]. Cada prova de resolução de um desafio precisaria ser assinada por duas chaves de desafio, uma do cliente e outra do servidor. A chave privada de desafio do cliente poderia ser derivada de uma semente gerada a partir da *flag*, exatamente como é hoje, mas a chave pública correspondente não seria divulgada. O par de chaves de desafio do servidor também seria mantido em sigilo. Apenas a chave pública combinada de cada desafio (agregando chave de cliente e de servidor) seria divulgada, permitindo assim a conferência da trilha de auditoria publicada pelo servidor. Pretende-se realizar provas de segurança, implementação e testes dessa alteração do protocolo em trabalhos futuros.

Por fim, além das perguntas já mencionadas, o questionário sugeriu diversas

possíveis mitigações para o problema do açambarcamento, avaliando-as em escala Likert [Likert 1932] (em parênteses, da forte aprovação à forte rejeição):

- dar bônus ao primeiro time que resolver um desafio (24%, 16%, 12%, 20%, 28%);
- após cada resolução, reduzir a pontuação do desafio apenas para equipes que ainda não o resolveram (16%, 16%, 12%, 16%, 40%);
- limitar taxa, número de submissões por tempo (12%, 20%, 12%, 20%, 36%);
- rotacionar *flags* (*e.g.* a cada 20 minutos) sempre que possível (4%, 24%, 16%, 32%, 24%);
- abolir *flags* convencionais, pedir aos competidores para executar um binário no servidor de cada desafio passando o nome do time como argumento (0%, 12%, 28%, 32%, 28%);
- esconder a pontuação e os desafios resolvidos, revelar aos times apenas sua posição no placar (12%, 4%, 4%, 36%, 44%);
- proibir o açambarcamento nas regras e deixar que os organizadores julguem casos suspeitos (24%, 16%, 24%, 16%, 20%).

Dentre as mitigações propostas, a que alcançou menor rejeição foi a última. Em campo de comentários aberto, alguns participantes sugeriram acrescentar a proibição ao açambarcamento nas regras, mas sem investigar casos suspeitos, apenas para deixar claro que a prática não é bem vista pela comunidade.

10. Conclusões

Este trabalho discutiu a evolução da plataforma utilizada para o Pwn2Win CTF partindo de uma plataforma implementada em PHP com MySQL em 2016, introduzindo o protocolo NIZKCTF em 2017 como ele foi proposto e melhorando sua implementação a cada ano, levando em consideração os comentários fornecidos pelos competidores depois de cada edição do evento. Demonstrou-se, assim, a dificuldade de coletar a real percepção dos usuários, pois comentários negativos relacionados ao uso do protocolo NIZKCTF cessaram somente em 2021, quando o cliente da plataforma foi implementado totalmente no navegador e independente de interação direta dos competidores com o GitHub. Desta forma, restaram críticas somente quanto a detalhes de implementação, tais como eficiência, que poderão ser mais facilmente melhorados nas próximas edições.

Com o uso do protocolo NIZKCTF e mitigados problemas mais fundamentais relacionados à plataforma em si, outra discussão veio a tona nas horas finais da edição de 2021 do evento, quando o time vencedor mostrou que estava usando a tática de açambarcamento de *flags*. Ao levantar essa pauta, a organização fez uma pesquisa quanto à opinião dos times com relação à prática e os resultados mostraram que a maioria dos times a consideram uma prática antidesportiva, mas não há consenso se a organização do evento deve ou não tentar mitigá-la. O protocolo NIZKCTF fornece publicamente para cada desafio uma chave pública que permite ao jogador saber se a *flag* é válida, de forma a permitir auditoria do resultado, porém isso permite aos times praticar o açambarcamento de *flags* sem risco de guardar uma *flag* incorreta. Em edições futuras, pretende-se impedir a validação *offline* de *flags* por meio de novas alterações no protocolo NIZKCTF.

Além disso, em trabalhos futuros pretende-se melhorar a forma como a trilha de auditoria é atestada. Na última versão da implementação, a atestação ocorre com a consulta do servidor por uma ferramenta externa em intervalos periódicos. Pretende-se subs-

tituir esse mecanismo por comunicação baseada em eventos, de forma que a atualização da trilha seja atestada o mais rápido possível após a submissão de uma *flag*.

Referências

Back, A. (2002). Hashcash - a denial of service counter-measure. http://www.hashcash.org/hashcash.pdf.

de Leon, D. C., Goes, C. E., Haney, M. A., and Krings, A. W. (2018). Adles: Specifying, deploying, and sharing hands-on cyber-exercises. *Computers & Security*, 74:12–40.

Karagiannis, S., Maragkos-Belmpas, E., and Magkos, E. (2020). An analysis and evaluation of open source capture the flag platforms as cybersecurity e-learning tools. In *IFIP World Conference on Information Security Education*, pages 61–77. Springer.

Kucek, S. and Leitner, M. (2020). An empirical survey of functions and configurations of open-source capture the flag (CTF) environments. *Journal of Network and Computer Applications*, 151:102470.

Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

Magalhães, L., Petri, A. C. F., Alves, G. d. S., Marcondes, C. A. C., and Matias, P. (2017). Provisionamento automatizado de servidores para competições de segurança da informação. In *Salão de Ferramentas do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, Brasília. SBC.

Matias, P., Barbosa, P., Cardoso, T. N., Campos, D. M., and Aranha, D. F. (2018). NIZKCTF: A noninteractive zero-knowledge capture-the-flag platform. *IEEE Security & Privacy*, 16(6):42–51.

Maxwell, G., Poelstra, A., Seurin, Y., and Wuille, P. (2019). Simple Schnorr multi-signatures with applications to Bitcoin. *Designs, Codes and Cryptography*, 87(9):2139–2164.

Mendonça, B. d. A. and Matias, P. (2021). Auditchain: a mechanism for ensuring logs integrity based on proof of existence in a public blockchain. In *2021 11th IFIP International Conference on New Technologies, Mobility and Security*, pages 1–5.

Senanayake, R., Porras, P., and Kaehler, J. (2019). Revolutionizing the visual design of capture the flag (CTF) competitions. In *International Conference on Human-Computer Interaction*, pages 339–352. Springer.

Taylor, C., Arias, P., Klopchic, J., Matarazzo, C., and Dube, E. (2017). CTF: State-of-the-art and building the next generation. In *2017 USENIX Workshop on Advances in Security Education*.

Vigna, G., Borgolte, K., Corbetta, J., Doupe, A., Fratantonio, Y., Invernizzi, L., Kirat, D., and Shoshitaishvili, Y. (2014). Ten years of iCTF: The good, the bad, and the ugly. In *2014 USENIX Summit on Gaming, Games, and Gamification in Security Education*.

Zhang, K., Dong, S., Zhu, G., Corporon, D., McMullan, T., and Barrera, S. (2013). picoCTF 2013-toaster wars: When interactive storytelling game meets the largest computer security competition. In *2013 IEEE International Games Innovation Conference*, pages 293–299. IEEE.