

ARTIGO CIENTÍFICO

ÁREA DE CONCENTRAÇÃO

**CIÊNCIA E
TECNOLOGIA**



GEOPOSICIONAMENTO DE SMARTPHONE NO C2 EM COMBATE 6.0: UM INTEGRADOR EM LINGUAGEM DE PROGRAMAÇÃO PYTHON

ADRIAN LIMA CORCINO DOS SANTOS¹, OSVALDO TENORIO VILELA DA COSTA²

Pós-graduado em Gestão de Sistemas Táticos de Comando e Controle¹, Pós-graduado em Gestão de Sistemas Táticos de Comando e Controle²

RESUMO: ESTE DOCUMENTO APRESENTA UMA SUGESTÃO DE DESENVOLVIMENTO DE APLICATIVO QUE SEJA CAPAZ DE GEOPOSICIONAR UM SMARTPHONE NO PROGRAMA C2 EM COMBATE 6.0, ASSIM COMO É REALIZADO NO SISTEMA PACIFICADOR. PARA ISSO FORAM REALIZADAS CAPTURAS DE TRÁFEGO DE REDE DO RÁDIO HARRIS 7800V-HH ENQUANTO TRANSMITIA INFORMAÇÕES DE COORDENADAS GEOGRÁFICAS PARA UM SERVIDOR EMULADO DO C2 EM COMBATE, AS QUAIS FORAM EXTRAÍDAS E ARMAZENADAS. EM SEGUIDA FOI DESENVOLVIDO UM SCRIPT EM LINGUAGEM DE PROGRAMAÇÃO PYTHON QUE CAPTURA AS COORDENADAS GEOGRÁFICAS REAIS DO SMARTPHONE E O ENVIA, NO MESMO FORMATO TRANSMITIDO PELO RÁDIO, PARA O SERVIDOR DE POSIÇÕES DO C2 EM COMBATE, QUE O DETECTA COMO UM EQUIPAMENTO RÁDIO ATIVO E GEOPOSICIONADO. ESTE ESTUDO TEVE COMO PROPÓSITO DEMONSTRAR A POSSIBILIDADE DE SE REALIZAR A MESMA FUNCIONALIDADE FORNECIDA PELO APLICATIVO PACIFICADOR MÓVEL, QUE ATÉ O MOMENTO INEXISTE OFICIALMENTE NO SISTEMA DO C2 EM COMBATE. CONCLUIU-SE, PORTANTO, QUE É POSSÍVEL REPLICAR A UTILIDADE DO APLICATIVO PACIFICADOR MÓVEL PARA OUTROS SISTEMAS DE COMANDO E CONTROLE DO EXÉRCITO, FORNECENDO UM MEIO DE ALTERNATIVO DE EQUIPAMENTO, ALÉM DO RÁDIO, ECONOMIZANDO RECURSOS E AMPLIANDO AS CAPACIDADES DE CONSCIÊNCIA SITUACIONAL DO SISTEMA.

PALAVRAS-CHAVE: COMANDO E CONTROLE, C2 EM COMBATE, PYTHON, GEOPOSICIONAMENTO.

INTRODUÇÃO

O presente trabalho trata sobre comando e controle, campo de pesquisa inserido na área de concentração Ciência e Tecnologia, conforme a Portaria nº 734, Art. 4º, inciso VI, de 19 de agosto de 2010, do Comandante do Exército.

Esta pesquisa tem como justificativa a inexistência de um aplicativo que integre dispositivos móveis ao software C2 em Combate, semelhante ao Pacificador Móvel no sistema Pacificador.

Com isso estabeleceu-se o objetivo de sugerir o desenvolvimento de um aplicativo capaz de, ao menos, geoposicionar smartphones no software C2 em Combate, ampliando as possibilidades de emprego e economizando equipamentos rádios.

Mais especificamente, este trabalho

demonstra que é possível interoperar entre o software e o dispositivo móvel, mesmo que não tenha sido projetado para isso.

Dessa forma, o escopo do tema visa contribuir com elementos de consciência situacional aplicados no software de comando e controle do Exército

Contudo, este trabalho se limita a fornecer uma aplicação simplificada ao propósito de geoposicionamento do dispositivo móvel e deixa aberta para futuras pesquisas que querem desenvolver a aplicação ou simplesmente incrementar funcionalidades ao código exposto.

1 METODOLOGIA

Este trabalho utilizou o método indutivo para formulação da pesquisa, iniciando-se por uma pesquisa documental e exploratória a



respeito da linguagem de programação em Python e suas aplicações

Após isso, foram feitas pesquisas laboratoriais onde foi emulado um servidor de C2 em Combate, versão 6.0, no programa VirtualBox e em seguida realizadas as capturas de tráfego de rede entre o Equipamento Rádio Harris 7800V-HH e o servidor.

Primeiramente, os dados coletados da captura foram analisados de forma a compreender efetivamente o que significava cada informação para que pudessem ser utilizadas posteriormente.

Após a completa compreensão dos dados, foi escrito um script em Python, para que os enviasse ao servidor de posições do C2 em Combate, simulando o Equipamento Rádio Harris 7800V-HH, e foi verificado que o servidor o reconhecia como um rádio legítimo e ativo.

Para atingir o objetivo de se replicar a funcionalidade do aplicativo Pacificador Móvel, o script foi adaptado para a plataforma Android em que, ao mesmo tempo que captura as informações geográficas do GPS do smartphone, as envia para o servidor do C2 em Combate, que também o detecta como um rádio e o geoposiciona em seu mapa.

2 DISCUSSÃO E RESULTADOS

2.1 C2 EM COMBATE

O programa C2 em Combate é um software de Comando e Controle pertencente ao Sistema Militar de Comando e Controle (SIS-MC²) (BRASIL, p. 20) que tem por finalidade fornecer dados em tempo real sobre as operações correntes provendo consciência situação e servindo de apoio à decisão às autoridades que comandam as operações.

Esse software é capaz de prover recursos de planejamento e gerenciamento de operações militares, por ser capaz de fornecer serviços de apresentação de mapas digitais, cartas topográficas ou imagens georreferen-

ciadas ao mesmo tempo que dispõem sobre eles instalações, unidades, veículos, pessoas e equipamentos de emprego militar.

Outra capacidade é que o C2Cmb fornece a possibilidade de interoperar com equipamentos rádios que enviam suas coordenadas geográficas e são dispostos, em tempo real, no mapa, além de ser capaz de emitir relatórios e documentos oficiais de operação.

Não obstante, o software é escalonado hierarquicamente, permitindo que informações sejam inseridas no contexto em que as frações de subordinadas possam replicá-las aos escalões superiores, que as visualizam e as filtram se assim desejarem.

Neste trabalho, foi criada uma máquina virtual com 80 Gb de HD e 3 Gb RAM e instalado o Windows 10 utilizando-se o software VirtualBox e, em seguida, instalado o servidor do C2 em Combate, versão 6.0064986, que funcionou perfeitamente.

2.2 CAPTURA DE TRÁFEGO DE REDE

Com a finalidade de entender como o rádio Harris 7800V-HH transmitia as coordenadas geográficas para o servidor de posições do C2 em Combate, foi utilizado o software Wireshark para realizar a leitura dos pacotes enviados.

Antes de realizar a captura, o rádio 7800V-HH foi renomeado com o nome de VITIMA, para que fosse mais fácil identificação, e assim foram capturados pacotes UDP (User Datagram Protocol) de destino do rádio para a máquina virtual do C2 em Combate.

2.2.1 User Datagram Protocol (UDP)

Segundo Neto (2017, p. 29), o UDP é um protocolo que não é orientado à conexão, pois ele não exige uma resposta de confirmação pelo receptor.

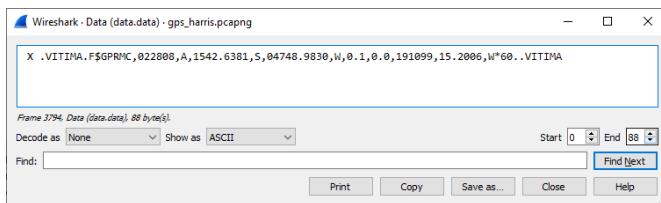
Esse protocolo é adequado para a transmissão de dados de GPS pois o contro-



le da transmissão é realizado pela aplicação do C2 em Combate, devido ao fato de que uma conexão TCP exigiria que as instâncias se mantivessem conectadas durante todo o processo, enquanto que a transmissão UDP se assemelha à característica de transmissão half-duplex do rádio, que exigem chaveamento do canal de transmissão e recepção.

2.2.1 Pacote UDP Capturado

FIGURA 1 Pacote Capturado pelo Wireshark.



Fonte: os autores, 2019

TABELA 1 Dados do pacote capturado.

DADOS	INFORMAÇÃO
X.	Início do pacote
VITIMA	Nome do rádio
.F	Não identificado
\$GPRMC	Identificador do padrão internacional NMEA (Especificação de dados de GPS mínima)
022808	Hora (No formato: HHMMSS H: Hora M: Minuto S: Segundo)
A	Aviso de navegação: Aviso de navegação: (No formato: A: Ok V: Aviso)
1542.6381	Latitude (No formato: GGMM.SS G: Graus M: Minutos S: Segundos em decimal)
S	Latitude (S: Sul, N: Norte)

04748.9830	Longitude (No formato: OGGMM.SS G: Graus M: Minutos S: Segundos em decimal)
W	Longitude (W: Oeste, E: Leste)
0.1	Velocidade de deslocamento
0.0	Azimute
191099	Data (No formato: DDMMAA D: Dia M: Mês A: Ano)
15.2006	Declinação Magnética
W	Orientação da Declinação (W: Oeste, E: Leste)
*60..	Checksum (hexadecimal)
VITIMA	Nome do rádio
,	Separador (vírgula)

Fonte: os autores, 2019.

2.2.3 Padrão internacional NMEA

Após pesquisas, foi identificado que as informações do pacote pertencem ao padrão internacional de formato de dados NMEA (National Marine Electronics Association), padrão, esse, desenvolvido visando a conexão de dispositivos eletrônicos marinhos (MundoGEO, 2004).

FIGURA 2 Padrão NMEA 0183

Fonte: MundoGEO <<https://mundogeo.com/blog/2004/01/01/gps-21-11/>> 2019

O padrão é constituído por até 82 caracteres ASCII, iniciando, sempre, com o símbolo \$, termina com o verificador de soma *checksum* e todas as informações são separadas por vírgulas.

Mais especificamente, o padrão utilizado pelo rádio segue o padrão de informações mínimas GPRMC (*Recommended Minimum Specific GPS Data*), que significa, recomendação de informações específicas mínimas de GPS. (BADDELEY, 2001)

FIGURA 3 GPRMC.

```

SGPRMC

Recommended minimum specific GPS/Transit data

eg1. $GPRMC,081836,A,3751.65,S,14507.36,E,000.0,360.0,130998,011.3,E*62
eg2. $GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68

225446      Time of fix 22:54:46 UTC
A             Navigation receiver warning A = OK, V = warning
4916.45,N   Latitude 49 deg. 16.45 min North
12311.12,W  Longitude 123 deg. 11.12 min West
000.5        Speed over ground, Knots
054.7        Course Made Good, True
191194       Date of fix 19 November 1994
020.3,E     Magnetic variation 20.3 deg East
*68          mandatory checksum

eg3. $GPRMC,220516,A,5133.82,N,00042.24,W,173.8,231.8,130694,004.2,W*70
    1   2   3   4   5   6   7   8   9   10  11  12

1  220516    Time Stamp
2  A          validity - A-ok, V-invalid
3  5133.82   Current Latitude
4  N          North/South
5  00042.24  current Longitude
6  W          East/West
7  173.8     Speed in knots
8  231.8     True course
9  130694    Date Stamp
10 004.2     Variation
11 W          East/West
12 *70       checksum

```

Fonte: Gleen Baddeley. <<http://aprs.gids.nl/nmea/#rmc>>, 2019.

Com as informações acima tornou-se possível reproduzir os pacotes no mesmo formato com informações obtidas de outro dispositivo, em nosso caso, com informações do GPS de um dispositivo móvel, somente sendo necessário adaptar as informações no formato adequado.

2.3 LINGUAGEM DE PROGRAMAÇÃO EM PYTHON

Python é uma linguagem de programação classificada como de altíssimo nível, de código aberto, disponível para diversos sistemas operacionais (Windows, Linux, Macintosh e Android) (NETO, 2007, p. 33) e interoperável

com diversas outras linguagens de programação (Java, PHP, R, C++, etc).

Essa linguagem não exige compilação para ser executada, só necessita de um interpretador que leia o código escrito, porém nada a impede de ser compilada caso necessário.

Para Neto (2007, p. 40), Python “[..] oferece excelentes mecanismos para modularizar o código-fonte.”, devido a sua capacidade de importar bibliotecas, ou módulos, que são conjuntos de código pré-escritos que realizam funcionalidade específicas de modo que o programador não tenha que recriar os códigos.

2.3.1 Biblioteca Socket

A biblioteca Socket é um conjunto de funções que permitem a comunicação entre máquinas na rede ou entre processos internos.

Neto (2007, p. 26) explica que sockets são utilizados para implementar aplicações que envolvem comunicações em redes TCP/IP e devem ser criados antes de criar sockets antes de iniciar qualquer tipo de comunicação.

Para reproduzir a comunicação do rádio com o servidor do C2 em combate, utilizamos o SOCK_DGRAM como tipo de socket para comunicação UDP.

Extraímos a parte do código que realiza a conexão e o envio das informações:

FIGURA 4 Conexão Socket UDP.

```

91 sockobj = socket(AF_INET, SOCK_DGRAM)
92 try:
93     resultado = sockobj.connect((self.destino, self.porta))
94     sockobj.send(gps.encode('utf-8'))
95 except:
96     continue
97     sockobj.close()

```

Fonte: Os autores, 2019.

No código acima é aberta um socket UDP (SOCK_DGRAM), em seguida, tentamos conexão com o ip e porta de destino e enviamos a informação de gps codificada em bytes no formato utf-8, após o envio a conexão é finalizada.



2.3.2 Biblioteca Kivy

A biblioteca Kivy é um framework para desenvolvimento multiplataforma escrito majoritariamente com a linguagem Python e, com alguns trechos escritos em Cython (Cython é um subconjunto da linguagem Python e seu objetivo é a conversão de código Python para código C “nativo”, Cython é o nome de uma linguagem e também, o nome do compilador). O framework permite o desenvolvimento de aplicações para os sistemas operacionais Microsoft Windows, Linux, Mac OSX, Android, iOS e Raspberry utilizando um mesmo código Python. O projeto é composto por vários subprojetos, cada um especializado numa determinada tarefa, como por exemplo, a geração de executáveis para determinada plataforma, uma API genérica para o fácil acesso ao hardware em plataformas diferentes, desenvolvimento de games etc.

O projeto Kivy, cujo site é <http://kivy.org>, é composto por vários subprojetos, dentre estes, a biblioteca Kivy (FILHO, 2018).

2.3.2.1 Plyer

Nessa pesquisa utilizou-se API (Application Programming Interface) Plyer para acesso ao recurso de geolocalização do sistema Android.

Plyer <<https://github.com/kivy/plyer>> é uma API de plataforma independente usado para acessar recursos comumente encontrados em várias plataformas, principalmente as plataformas móveis. A ideia é fazer com que o seu aplicativo possa invocar funções simplesmente Plyer, seja para apresentar uma notificação ao usuário, enquanto o Plyer cuidará de como conversar com a plataforma corretamente, e isso tudo, independentemente da plataforma ou sistema operacional (EXCRIPT, [2018?]).

Nessa pesquisa utilizou-se como base, para desenvolvimento do aplicativo de geolocalização, um script da GitHub, desenvolvido na linguagem python com Kivy e Plyer, esse

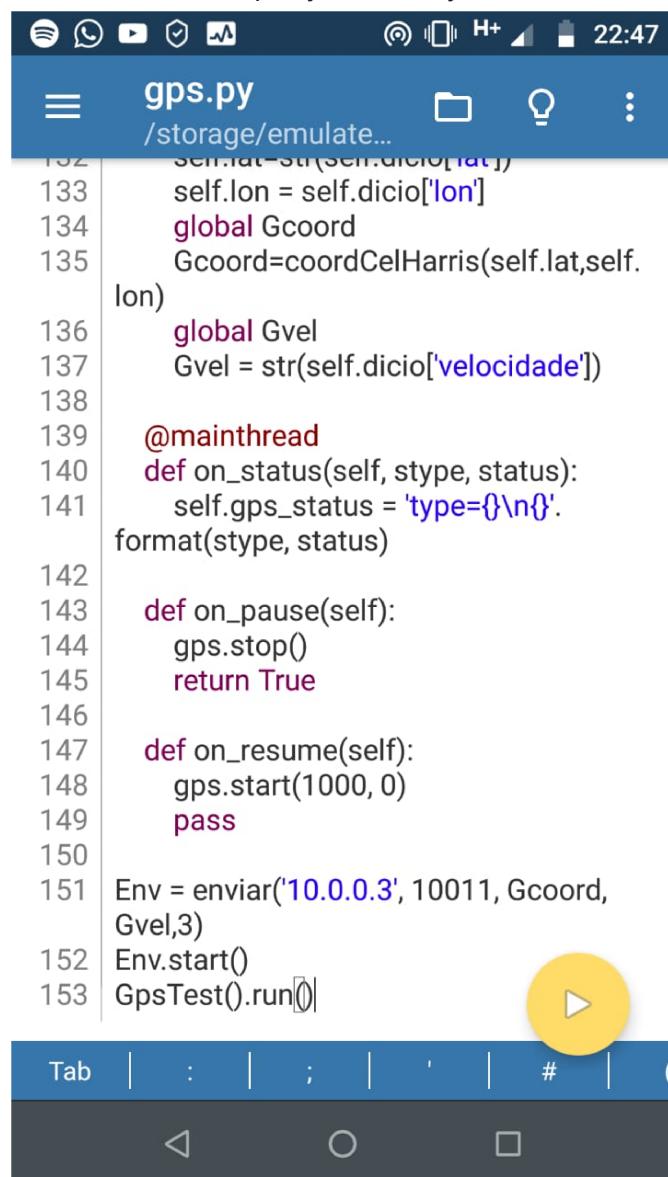
aplicativo, basicamente tem a função de exibir as coordenadas geográficas na tela de um smartphone (GITHUB, 2017).

2.4 GEOPOSICIONAMENTO DO SMARTPHONE

Para executarmos o script de geoposicionamento no Android foi necessário instalar um aplicativo interpretador de Python no smartphone.

Baixamos e instalamos o Pydroid3 e o Pydroid *Permission Plugin* para permitir ao aplicativo o acesso as informações de gps do smartphone, com isso conseguimos executar o script sem problemas.

FIGURA 5 Script Python no Pydroid.

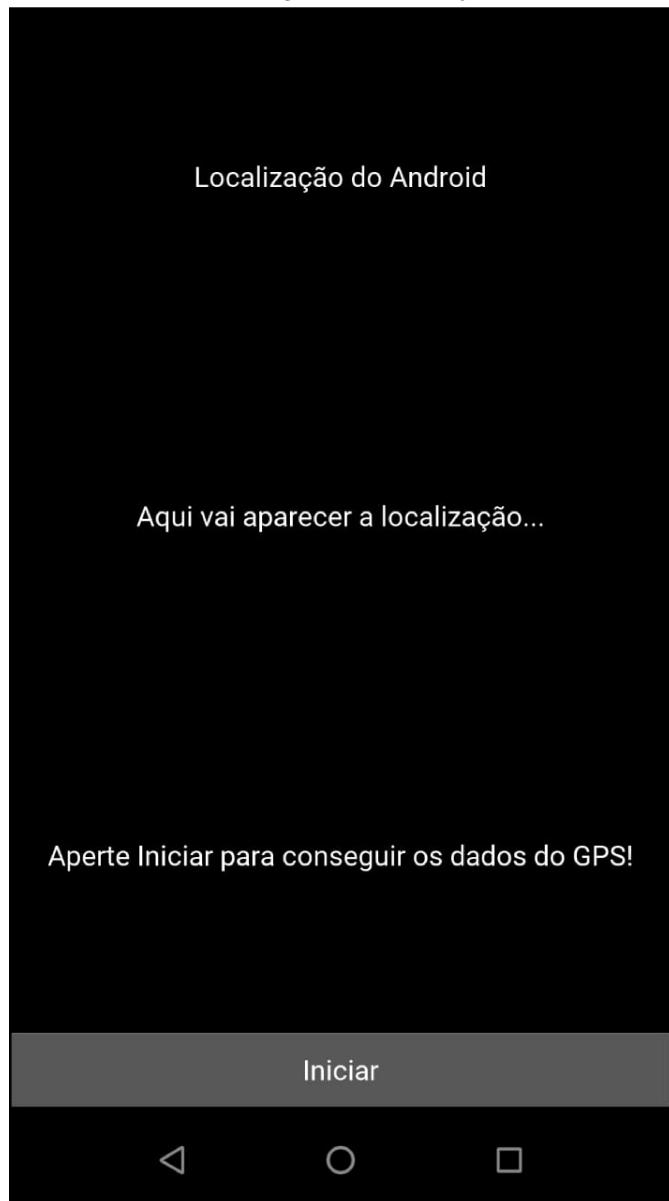


```
gps.py
/storage/emulate...
132 self.lat=Str(self.dicio['lat'])
133 self.lon = self.dicio['lon']
134 global Gcoord
135 Gcoord=coordCelHarris(self.lat,self.
136 lon)
137 global Gvel
138 Gvel = str(self.dicio['velocidade'])
139 @mainthread
140 def on_status(self, stype, status):
141     self.gps_status = 'type={}\n{}'.format(stype, status)
142
143     def on_pause(self):
144         gps.stop()
145         return True
146
147     def on_resume(self):
148         gps.start(1000, 0)
149         pass
150
151 Env = enviar('10.0.0.3', 10011, Gcoord,
152 Gvel,3)
153 Env.start()
GpsTest().run()
```

Fonte: Os autores, 2019.

Ao executarmos o script é iniciada a instância gráfica a qual é necessário pressionar o botão iniciar para que os dados sejam carregados e enviados.

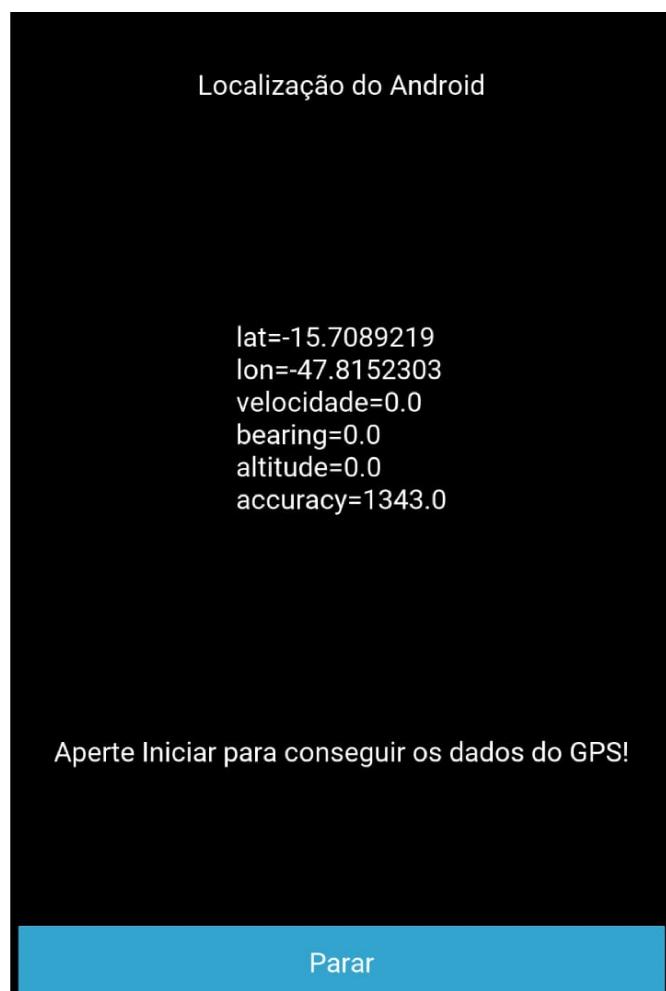
FIGURA 6 Aplicação Kivy iniciada.



Fonte: Os autores, 2019.

Ao pressionarmos o botão iniciar são carregadas as informações de gps do smartphone e inicia-se o envio para o ip do servidor de C2 em Combate configurado. O servidor do C2 em Combate detecta efetivamente as informações como um rádio e projeta a visualização do objeto geoposicionado com a localização exata do dispositivo móvel, confirmando a funcionalidade do código.

FIGURA 7 Dados de GPS e envio



Fonte: Os autores, 2019.

CONCLUSÃO

Diante do exposto verificamos que a execução do código foi capaz de cumprir o objetivo de enviar a localização do smartphone para o servidor de C2 em combate, reproduzindo a funcionalidade do seu semelhante, o Pacificador Móvel.

Essa simples funcionalidade soluciona o problema inicial dessa pesquisa que é a inexistência de um aplicativo que integre dispositivos móveis ao software C2 em Combate e possibilita a economia de equipamentos rádio e flexibiliza o uso em localidades onde o militar sai do alcance da rede, sendo possível operar em áreas com cobertura de telefonia móvel em dados e acesso VPN à rede EBNet.

Com isso o usuário do smartphone, reporta sua localização em tempo real e tem a possibilidade de acessar o C2 em combate



```

    item->FirstChildElement(); item := []: item < item->FirstChildElement();
    item->Attribute( "name" );
    item->Attribute( "type" );
}

Item_name = item->Attribute( "name" );
SpriteName = item->Attribute( "spritename" );
ost::lexical_cast<float>( item->Attribute("x") );
ost::lexical_cast<float>( item->Attribute("y") );
z = boost::lexical_cast<float>( item->Attribute("z") );

st::iterator sp = sprite_descs.begin();
sprite_descs.end(); ++sp )

```

pelo navegador, o que o permite lançar informações dos incidentes ocorridos. Isso pode ser uma melhora no conceito de emprego do C2 em combate, pois dessa forma não é necessário que o militar no terreno faça um relatório para o COp para que os operadores do COp reportem as informações pelo C2 em Combate ao escalão superior, mas a informação é lançada no programa diretamente do militar no terreno, de forma semelhante ao Pacificador móvel. A desvantagem é que o C2 em Combate não possui página responsiva que se adapta ao tamanho da tela, o que dificulta a operação pelo smartphone.

O escopo do trabalho se limitou ao geoposicionamento de dispositivo móvel, mas existem diversas aplicações que podem ser exploradas utilizando-se das informações descritas nesta obra.

Com isso, o trabalho deixa de legado o código fonte para futuras pesquisas que desejem aprimorá-lo, ou mesmo, de base para de-

senvolvimento em outras linguagens e incremento de novas funcionalidades ao sistema.

SMARTPHONE GEO-POSITIONING IN C2 EM COMBATE 6.0 SOFTWARE: AN INTEGRATOR IN PYTHON PROGRAMMING LANGUAGE

ABSTRACT: THIS DOCUMENT PRESENTS A SUGGESTION OF APPLICATION DEVELOPMENT THAT IS ABLE TO GEOFACILITATE A SMARTPHONE IN THE PROGRAM C2 EM COMBATE 6.0, AS IT IS CARRIED OUT IN THE PACIFICADOR SYSTEM. FOR THIS PURPOSE, NETWORK TRAFFIC CAPTURES OF THE HARRIS 7800V-HH RADIO WHILE TRANSMITTING GEOGRAPHIC COORDINATE INFORMATION TO A EMULATED SERVER OF C2 EM COMBATE, WHICH WERE EXTRACTED AND STORED. NEXT, A SCRIPT WAS DEVELOPED IN PYTHON PROGRAMMING LANGUAGE THAT CAPTURES THE REAL GEOGRAPHICAL COORDINATES OF THE SMARTPHONE AND SENDS IT, IN THE SAME FORMAT TRANSMITTED BY THE RADIO, TO THE POSITION SERVER OF THE C2 EM COMBATE, WHICH DETECTS IT AS ACTIVE. THIS STUDY AIMED TO DEMONSTRATE THE POSSIBILITY OF PERFORMING THE SAME FUNCTIONALITY PROVIDED BY THE PACIFICADOR MÓVEL APPLICA-



TION, WHICH HAS NOT OFFICIALLY EXISTED IN THE C2 EM COMBATE SYSTEM. IT WAS THEREFORE CONCLUDED THAT IT IS POSSIBLE TO REPLICATE THE USEFULNESS OF THE PACIFICADOR MÓVEL APPLICATION TO OTHER COMMAND AND CONTROL SYSTEMS BY PROVIDING AN ALTERNATIVE MEANS OF EQUIPMENT IN ADDITION TO THE RADIO, SAVING RESOURCES AND ENHANCING THE SITUATIONAL AWARENESS CAPABILITIES OF THE SYSTEM .

KEYWORDS: COMMAND AND CONTROL, C2 EM COMBATE, PYTHON, GEO-POSITIONING.

REFERÊNCIAS

BADDELEY, Gleem. GPS - NMEA sentence information. 2001. Disponível em: <<http://aprs.gids.nl/nmea/#rmc>>. Acesso em: 19 jun. 2019.

BRASIL, ESTADO-MAIOR DO EXÉRCITO; Compreensão das Operações (COMOP) nº 01/2019, Apoio de Comunicações à Força Terrestre. Portaria nº 023-EME, de 31 de janeiro de 2019. Brasília, DF, 2019. Disponível em: <<http://www.sgex.eb.mil.br/sistemas/be/copiar.php?codarquivo=1662&act=bre>>. Acesso em: 17 jun. 2019.

EXCRIPT. Kivy: Documentação. [2018?] Disponível em: <http://excript.com/downloads/kivy-pt_br-excript.pdf>. Acesso em: 16 jun. 2019.

FILHO, Cláudio Rogério Carvalho. Biblioteca kivy. Excript, 21 jun. 2018. Disponível em: <<http://excript.com/python/kivy.html>>. Acesso em: 16 jun. 2019.

GITHUB. 2017. Disponível em:<<https://github.com/kivy/plyer/blob/master/examples/gps/main.py>>. Acesso em: 16 jun. 2019

MUNDOGEO. GPS 21. 2004. Disponível em: <<https://mundogeo.com/blog/2004/01/01/gps-21-11/>>. Acesso em: 19 jun. 2019.

NETO, Jahyr Gonçalves. Desenvolvimento de uma Plataforma Multimídia Utilizando a Linguagem Python. 2007. p. 89. Dissertação de mestrado (Engenharia Elétrica). Disponível em: <http://repositorio.unicamp.br/jspui/bitstream/REPOSIP/259651/1/GoncalvesNeto_Jahyr_M.pdf>. Acesso em: 16 jun. 2019.

PYTHON. Disponível em: <<https://www.python.org/about/>>. Acesso em: 16 jun. 2019.

SEITZ, Justin. Black Hat Python: Programação Python para hackers e pentesters. 1. ed. São Paulo: Novatec Editora, 2015. ISBN 978-85-7522-420-5.

WIRESHARK. Disponível em: <<https://www.wireshark.org/index.html#aboutWS>>. Acesso em: 16 jun. 2019.

Os autores são bacharéis em Ciências Militares pela Academia Militar das Agulhas Negras (AMAN) e são pós-graduandos em Gestão de Sistemas Táticos de Comando e Controle pela Escola de Comunicações.

Atualmente, o 1º Ten Adrian exerce a função de Chefe da 3ª Seção da 8ª Companhia de Comunicações, Bento Gonçalves-RS e pode ser contactado pelo e-mail: adrian.santos@eb.mil.br.

Atualmente, o 1º Ten Tenorio exerce a função de Chefe da Seção de Tecnologia da Informação do 9º Batalhão de Comunicações e Guerra Eletrônica, Campo Grande-MS e pode ser contactado pelo e-mail: tenorio.costa@eb.mil.br.

